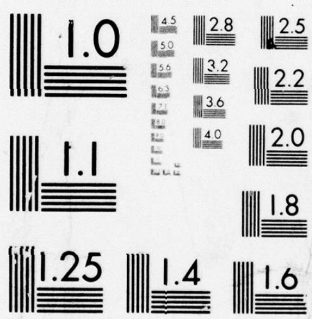
AD-A036 346 GEORGIA UNIV ATHENS DEPT OF STATISTICS AND COMPUTER--ETC F/G 9/2
GRAPHICAL AIDS FOR STATISTICAL COMPUTATION.(U)
DEC 76 W P BOND, R E BARGMANN N00014-69-A-0423
UNCLASSIFIED TR-112 NL

1 OF 4
AD
A036346


1.0
4.5
5.0
5.6
6.3
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

(5)

5

TECHNICAL REPORT NUMBER 112

THEMIS REPORT NUMBER 37

GRAPHICAL AIDS FOR STATISTICAL COMPUTATION

BY

Walter P. Bond and Rolf E. Bargmann

See 1473 in back

Reproduction in whole or in part is permitted for any purpose of the United States Government. This research was supported by the Office of Naval Research, Contract No. N00014-69-A-0423, NR 042-261.

Rolf E. Bargmann
Principal Investigator

The University of Georgia
Department of Statistics and Computer Science
Athens, Georgia

December, 1976

TABLE OF CONTENTS

LIST OF EXAMPLES

CHAPTER I

INTRODUCTION

In many instances it would be of great benefit to the statistician to be able to view graphically a mathematical surface with which he is working. A frequent occurrence of this circumstance is found in the exploration and optimization of response surfaces. Closely related to this problem is the maximization of likelihood surfaces. A second, equally important need can be filled by the ability to examine the behavior of statistical distribution functions whose parameters have been subjected to certain conditions.

In the past, it has been necessary for the statistician to resort to tedious manual plotting methods; however, modern technology has provided all of the tools necessary to accomplish the task: the computer, the plotter and the graphical computer terminal. Unfortunately, the statistician encounters many obstacles in attempting to take advantage of these devices because of the complexity involved in bringing them into use. The statistician has neither the time nor the inclination to interrupt his analysis to

1. identify the graphics equipment available to him,
2. secure documentation for the plotting software,
3. learn to use this software and
4. write and debug the specific computer program required for displaying the mathematical surfaces involved in his particular problem.

In most situations the statistician elects to ignore the possible use of graphics and to either rely on intuition as to the nature of the surface in question or to proceed with classical numerical techniques which have been developed to circumvent the inability to visualize the surface. The first of these solutions is likely to produce inaccurate results due to the lack of experience on the part of the statistician in examining actual surfaces. The second general method is always costly and can lead to erroneous results if there has been a poor choice of initial conditions.

A rather specialized approach to providing statisticians with interactive graphics capabilities was made by Ball and Hall [2]. Their system, PROMENADE, was designed primarily for clustering and pattern recognition. The authors point out, however, "that no limited approach to the problem will be as fruitful as a more integrated approach." The purpose of the following work is to indicate an "integrated approach"

which provides the statistician with easily accessible graphical aids with a minimum of emphasis upon the technical requirements imposed by the computer. The direction of the research has been toward the development of interactive graphics programs which allow the user to specify, in some simple fashion, the surface of interest, and which will then calculate <u>only once</u> an interpolation function for the surface. The user may then employ this interpolation function repeatedly, and cheaply, to display those portions of the surface under investigation.

To this end, two programs have been developed under the Graphics Monitor System (GMS) currently operated by the Department of Statistics and Computer Science at the University of Georgia. Penn [17] provides a complete description of the hardware and software characteristics of the system.

The first of the programs, YATES, analyzes data obtained from a factorial experiment. After the initial input of the data, the program performs an analysis of variance in order to test for significant main effects and interactions and, consequently, calculates the response equation, deleting, if desired, the coefficients of orthogonal polynomials corresponding to those effects which do not appear to be significant. This resulting mixed polynomial may be displayed in a number of ways, e.g., the equation itself may be viewed, cross-sections of

the surface may be plotted (with or without 95% confidence intervals) and, finally, surface contours may be generated. An interactive computer approach to the design and analysis of factorial experiments is demonstrated by Margolin [14] using interactive computing in the APL language. His methods, however, require a knowledge of either APL or an equally sophisticated vector oriented programming language.

The method, first introduced by Yates [3] for the analysis of 2^k factorial experiments, employed by the YATES program is an extension of simple polynomial interpolation to multidimension space. An additional discussion of this technique can be found in Davies [7]. Generalizations of this method to complete factorial experiments of higher order can be found in Binet, Leslie and Radock [4], Margolin [14], and Cooper [6]. Cooper discusses extending the method to accommodate 4 or more levels; however, no computational details are given. An additional "Yates-type" algorithm is introduced by Hunter [11] for the prediction of a single point on the response surface directly from the coefficients of the orthogonal polynomials. This technique is referred to by Hunter as the "Inverse Yates Algorithm." However, the technique is presented only for the 2^k case and, further, can only be used to predict values for the settings at which the responses were measured, i.e., it does not allow for

interpolation. The "Back Solution" used by the present program is a more general tool; one which reparametrizes the response equation in terms of orthogonal polynomials into an equation in terms of the original polynomial products. Chapter II, section 1, supplies the mathematical procedures necessary for the development and understanding of the Yates method. Chapter III, sections 1 and 3, describe the actual computer implementation of the techniques. Illustrations of some of the uses to which this general response surface analysis program may be put, e.g., analysis of laboratory experiments, problems in process evaluation and the investigation of statistical likelihood functions, are found in Chapter IV.

The second program, BLOWUP, resulted from the addition of plotting capability to the existing GMS calculator mode, CALCG [17]. A plot created by this unit consists of eleven equally spaced realizations of any function available in CALCG with intermediate values supplied by a spline interpolation procedure. Spline interpolation was introduced by Schoenberg [2]. General theory and applications of the spline function have been further developed by Ahlberg, Nilson and Walsh [1]. An elementary discussion of the subject can be found in Pennington [18]. Several researchers in the field of statistics are investigating the data fitting possibilities of splines; specifically, Poirier [19] and Wold [30] are researching the application

of cubic splines to problems of piecewise regression. The present program, however, utilizes the spline function in its original form as a device for numerical interpolation. Chapter II, section 2, contains the mathematical development of the spline algorithm. Chapter III, sections 2 and 4, discuss the computer implementation of the technique.

Examples illustrating the importance of the method are found in Chapter V, which discusses several important statistical problems which are made more tractable through this combination of the CALCG function and the spline interpolation technique. For example, the ability to calculate the power of the F test plays an important part in the design and analysis of experiments. It is necessary in order to answer *a priori questions* concerning the choice of sample size, the number of treatments, etc. It is convenient, after the fact, to be able to evaluate the sensitivity of the test performed. Because of the computational complexity of the noncentral F distribution and the related noncentral Beta distribution as described in Bargmann [3] and Thomas [25], statisticians have been limited to tables published in 1938 by Tang [24] and charts published in 195[illegible] by Pearson and Hartly [16] and in 1956 by Fox [8]. In 1967, Tiku [26] produced a more comprehensive set of tables. In 1972, Tiku [27] extended his earlier tables to include α=.10. However, the enormous variation of degrees of freedom, noncentrality parameters and α levels limits the

practical use of any set of tables of the noncentral F distribution. An attempted solution to this problem involving simplication of the power formulas was presented by Wheeler [29] in 1974 under the title of "Portable Poer". The pitfalls in this approach were immediately pointed out by Bowman and Kastenbaum [5]. The present program gives the statistician immediate access to the non-central Beta distribution and thus allows the use of the power of the F-test as an easily accessible tool.

The organization of the remaining chapters of this work is modular in nature. Chapter II describes the mathematical procedures which have been included in the two programs described above. The actual computer implementation of these programs is explained in Chapter III. Chapters IV and V contain step-by-step descriptions of actual interactive sessions utilizing the YATES and BLOWUP programs, respectively. Chapter VI summarizes the results of the research. Specific information concerning the technique of orthogonalization is found in Appendix A, while the actual orthogonal polynomials used in the YATES program are described in Appendix B. An example of the actual application of the Yates method is included in Appendix C. Program listings can be found in Appendix D and E.

CHAPTER II

MATHEMATICAL PROCEDURES

Certain mathematical procedures must be employed to produce the software modules introduced in Chapter I. The intent of this chapter is to describe the relevant procedures.

2.1 Mathematical Procedures for Fitting Tensor Products of Polynomials (Extended Yates Method)

The factorial experiment provides the statistician with a technique which allows him to assess the effect on the output or yield of a process brought on by the varying of experimental conditions over certain predefined ranges. Each experimental condition which is to be varied is referred to as a factor, e.g., temperature. Each setting over which the factor is to be varied is known as a level e.g., level 1 is 70°, level 2 is 80°, level 3 is 100°.

Each observation taken at a given factor combination is referred to as a response. The analysis of data resulting from a factorial experiment is performed for the purpose of estimating the main effects and interactions of the factors and, also, for calculating the associated

mean squares which are needed in order to determine the statistical significance of these effects. In addition, the experimenter who is concerned with determining the particular combination of factor levels which is expected to produce an optimum yield will wish to estimate the response function: $E[Y]=G(z_1, z_2, \ldots z_k)$ for which Y is the response variable and the z_i is a particular value setting for the i^{th} factor. This response function is usually represented by a polynomial in the z_i's.

A computational technique to aid in the analysis of data from a factorial experiment has been presented by Yates [3]. This method, originally presented for factors at two levels only, involves the reparametrization of the response equation from its form of a polynomial in the z_i's into a function of certain orthogonal polynomials $\Phi_j(z)$. Appendix A contains an explanation of the orthogonalization technique and Appendix B presents the set of orthogonal polynomials used in the present work. If the response data are arranged in a given well-defined order (to be explained in this chapter), then a series of repeated specified linear combinations of the data will result in, not only the estimates of the main effects and interactions along with their associated single degree of freedom sums of squares, but also, in the estimates of the coefficients of the orthogonal polynomials for the re-

parametrized response function. It is the purpose of the following sections to discuss the specific mathematical procedures necessary for the application of the Yates method. Appendix C contains an example of the method.

2.1 Raw Values and Standard Values

It can be seen in Appendix B that the set of orthogonal polynomials requires the level settings, z, to assume standardized values, viz,

2 levels

$z=-1$ for low $\quad$ $z=1$ for high

3 levels

$z=-1$ for low $\quad$ $z=0$ for middle $\quad$ $z=1$ for high

4 levels

$z=-1$ for very low	$z=-1/3$ for low	$z=1/3$ for high	$z=1$ for very high

The response equation produced is in the form a tensor product of polynomials where each of the input variables (factors) is assumed to be in the standard form (z-values). Thus, the values are dimension free, and the coefficients have the same dimension as the response output variable.

Due to the fact that the actual level settings or raw values, T, are measured in units corresponding to the

physical factors in question, e.g., temperature in degrees, it is necessary to find a transformation which will allow us to convert the original (raw) values into standard values. Similarly, an inverse transformation from the standard values back into the original values is also required. The remainder of this section presents these transformations.

Two Levels

The transformation from original (raw) values into standard values is always linear for 2 levels. Let T denote a value of the input variable, T_L the value corresponding to the low level, T_H the value corresponding to the high level, then

$$z = -1 + \frac{2}{T_H - T_L}(T - T_L) \qquad z = -1, +1$$

$$T = \frac{T_H + T_L}{2} + \frac{T_H - T_L}{2} z$$

Three Levels

For three levels, the transformation is linear only if the raw values corresponding to low, medium and high are equally spaced. Otherwise, a quadratic transformation is required:

$$z = -1 + \frac{2}{T_H - T_L}(T - T_L)$$

$$- \frac{2T_M - T_H - T_L}{(T_M - T_L)(T_H - T_L)(T_H - T_M)}(T - T_L)(T_H - T)$$

$$z = -1,0,1$$

$$T = T_M + \frac{T_H - T_L}{2} z + \frac{T_H + T_L - 2T_M}{2} z^2$$

Four Levels

For four levels, the formulas may have terms up to the third degree; as in the three level case, the transformation is linear if the raw observations T_{LL} (very low), T_L (low), T_H (high) and T_{HH} (very high) are equally spaced. To obtain T (raw observation) from z (a value interpolated in the standard scale) we can use the formula:

$$T = \frac{-T_{LL} + 9T_L + 9T_H - T_{HH}}{16}$$

$$+ \frac{T_{LL} - 27T_L + 27T_H - T_{HH}}{16} z$$

$$+ \frac{9(T_{LL} - T_L - T_H + T_{HH})}{16} z^2$$

$$+ \frac{9(3T_L - T_{LL} - 3T_H + T_{HH})}{16} z^3$$

$$z = -1.-1/3,1/3,1$$

A closed expression for the general (cubic) conversion from raw values (T) into standard values (Z) is more effectively performed by the familiar Newton divided-

difference scheme:

x	y	f[,]	f[,,]	f[,,,]
T_{LL}	-1			
T_L	-1/3	a		
T_H	1/3	*	b	
T_{HH}	1	*	*	c

$$z = ((c(T - T_H) + b)(T - T_L) + a)(T - T_{LL}) - 1$$

More Than Four Levels

Due to the infrequent use of factorial designs which involve factor settings at more than four levels, the present work does not consider these cases, However, although the resulting mathematical expressions are somewhat involved, a direct extension of the methods presented in this section could be made.

2.1.2 Generation of the Yates Order

The introduction to the current section contains a reference to the Yates method for ordering the response data. This technique can be executed symbolically as follows: For a given factor, A, the absence of the letter indicates the lowest level of the factor, while the presence of the letter a indicates the next higher level; a^2, the next higher level, and so on. For example, the Yates order for three factors A, B and C with levels 2, 3

and 2, respectively, is generated symbolically as follows:

The Yates Order
for Factors
A(2 levels) B(3 levels) C(2 levels)

Response Number	Symbolic Representation	Corresponding Levels A	B	C	Corresponding z Values z_A	z_B	z_C
1	(1)	1	1	1	-1	-1	-1
2	a	2	1	1	1	-1	-1
3	b	1	2	1	-1	0	-1
4	ab	2	2	1	1	0	-1
5	b^2	1	3	1	-1	1	-1
6	ab^2	2	3	1	1	1	-1
7	c	1	1	2	-1	-1	1
8	ac	2	1	2	1	-1	1
9	bc	1	2	2	-1	0	1
10	abc	2	2	2	-1	-1	1
11	b^2c	1	3	2	-1	-1	1
12	ab^2c	2	3	2	1	-1	1

It is the purpose of the section to review the modulo arithmetic necessary to associate a given response location with its unique combination of factor settings and, inversely, to determine, from a given combination of factor settings, the unique Yates location of the associated response. For simplicity, level values, rather than z

values, are used in this exposition.

Given a factorial design with n factors whose corresponding levels are $\ell_1, \ell_2, \ldots \ell_n$, respectively, the factor setting (1 for level 1, 2 for level 2, etc.) for the k<u>th</u> factor of the i<u>th</u> observation in the Yates order may be found by

$$j_1 = m + 1$$

$$j_k = [m / \prod_{\nu=1}^{k-1} \ell_\nu] + 1$$

where $m = (i-1)\bmod(\prod_{\nu=1}^{k} \ell_\nu)$

<u>Example</u> 2.1.2.1

Consider a 3x4x3x2 factorial design. The Yates ordering of the factor settings for the 60<u>th</u> observation is found by

factor 1 (k=1)

$m = 59\bmod 3 = 2$ and $j_1 = 2 + 1 = 3$

factor 2 (k=2)

$m = 59\bmod(3x4) = 11$ and $j_2 = 11/3 + 1 = 4$

factor 3 (k=3)

$m = 59\bmod(3x4x3) = 23$ and $j_3 = 23/12 + 1 = 2$

factor 4

$m = 59\bmod(3x4x3x2) = 59$ and $j_4 = 59/36 + 1 = 2$

Therefore, the Yates ordering of the factor settings for the 60th observation is

obs. no.	factor 1	factor 2	factor 3	factor 4
60	3	4	2	2

The process may be inverted by the following

$$i = \sum_{k=2}^{n} (j_k - 1) \prod_{\nu=1}^{k-1} \ell_\nu + j_1$$

Example 2.1.2.2

Consider the above 3x4x3x2 factorial design. The Yates order factor settings 3, 4, 2 and 2, respectively, yield

$$\begin{aligned} i &= (j_2-1)\ell_1 + (j_3-1)\ell_1\ell_2 + (j_4-1)\ell_1\ell_2\ell_3 + j_1 \\ &= (((j_4-1)\ell_3 + j_3-1)\ell_2 + \ell_2-1)\ell_1 + j_1 \\ &= (((2-1)3 + 2 - 1)4 + 4-1)3 + 3 \\ &= 60 \end{aligned}$$

So that factor settings 3, 4, 2 and 2 correspond to the 60th observation. Note that an easy computational form of the algorithm is given in the second line of the above example.

2.1.3 Application of the Yates Algorithm

In order to calculate the estimates of the coeffi-

cients of the orthogonal polynomial formulation of the general linear model as described above, certain linear combinations of the observations must be calculated. The calculations are greatly simplified by the application of the Yates algorithm which is composed of

a. a specific ordering of the observations and
b. the formulation of stepwise linear combinations of these ordered observations.

The procedure is applied by first totaling the replications for each factor combination (cell) and then arranging these subtotals into a vector using the Yates ordering (refer to section 2.1.2). Beginning with this vector, a new vector is created for each factor in the design; each vector, composed of linear combinations of the elements of the previous vector. The linear combinations are created sequentially by making one complete "pass" through the vector for each level of the current factor, according to the following scheme:

no. of levels	movement	pass	linear combination
2	pairwise	1	H + L
		2	H - L
3	triples	1	H + M + L
		2	H - L
		3	H + L - 2M

no. of levels	movement	pass	linear combination
4	quadruples	1	HH + H + L + LL
		2	3(HH - LL) + (H - L)
		3	(HH + LL) - (H + L)
		4	(HH - LL) - 3(H - L)

where

LL - very low, L - low, M - middle, H - high, HH - very high.

To obtain the coefficients of the response equation (in terms of the orthogonal polynomials), the elements of the resulting vector may then be divided by the products of the number of replications of the experiment and the sums of squares of the orthogonal polynomials (refer to Appendix C) corresponding to the treatment levels represented by the position of the element within the vector. In addition, the usual single degree of freedom sums of squares may be obtained by first squaring the elements of the resulting vector prior to the indicated division.

2.1.4 The Back Solution

It was stated in the previous section that the Yates method results in the calculation of estimates of the coefficients of the orthogonal polynomial formulation of the response function. In the simple case of one factor, at, say, 3 levels, a typical result might be

$$\hat{y} = 10 - 2\Phi_1(z) + 4\Phi_2(z)$$

where $\Phi_1(z)$ and $\Phi_2(z)$ are defined in Appendix B. It is reasonable that the experimenter may wish to reformulate the expression in terms of the standard z values.

$$\hat{y} = 10 - 2(z) + 4(3z^2-2)$$
$$= 2 - 2z + 12z^2$$

Because z lies between -1 and 1, the contribution of each coefficient to the predicted value of y may be readily determined by simple inspection of the resulting polynomial.

For the one factor case at 2, 3 or 4 levels, it is a straight forward procedure to substitute for the Φ's and collect terms in order to determine the rule necessary to reparamatrize the orthogonal formulation back to the standard form of the polynomial. The manipulation and resulting rule follows for each case. The notation followed in stating the rule is identical to that used in the previous section; however, in this instance, it is being applied to the $\hat{\beta}$'s.

Linear

$$\hat{y}=\hat{\beta}_0+\hat{\beta}_1\Phi_1(z)$$
$$\hat{y}=\hat{\beta}_0+\hat{\beta}_1 z$$

$$\hat{\alpha}_0=\beta_0 \text{ and } \hat{\alpha}_1=\beta_1$$

Rule: Copy L, Copy H

Quadratic

$$\hat{y}=\hat{\beta}_0+\hat{\beta}_1\Phi_1(z)+\hat{\beta}_2\Phi_2(z)$$

$$\hat{y}=\hat{\beta}_0+\hat{\beta}_1 z+\hat{\beta}_2(3z^2-2)$$

$$=(\hat{\beta}_0-2\hat{\beta}_2)+\hat{\beta}_1 z+3\hat{\beta}_2 z^2$$

$$\hat{\alpha}_0=\beta_0-2\hat{\beta}_2 \qquad \alpha_1=\hat{\beta}_1 \qquad \alpha_2=3\hat{\beta}_2$$

Rule: L-2H, copy M, 3H

Cubic

$$\hat{y}=\hat{\beta}_0+\hat{\beta}_1\Phi_1(z)+\hat{\beta}_2\Phi_2(z)+\hat{\beta}_3\Phi_3(z)$$

$$=\hat{\beta}_0+\hat{\beta}_1+\hat{\beta}_2(9/4z^2-5/4)+\hat{\beta}_3(45/4z^3-41/4z)$$

$$=(\hat{\beta}_0-5/4\beta_2)+3\hat{\beta}_1-41/4\hat{\beta}_3)z+9/4\hat{\beta}_2 z^2+45/4\hat{\beta}_3 z^3$$

$$\hat{\alpha}_0=\hat{\beta}_0-5/4\hat{\beta}_2, \hat{\alpha}_1=3\hat{\beta}_1-41/4\hat{\beta}_3, \hat{\alpha}_3=9/4\hat{\beta}_2, \hat{\alpha}_3=45/4\hat{\beta}_3$$

Rule: 12(LL)-15(H), 36(L)-123(HH), 27(H), 135(HH)

and divide all results by 12.

The technique developed above may be extended to the two factor case. Assume a 2x3 factorial design with the 2-level factor occurring first in the Yates ordering. Assume, further, that resulting estimated response function is

$$\hat{y}=5-2\Phi_1^{(1)}(z_1)+3\Phi_1^{(2)}(z_2)+5\Phi_1^{(1)}(z_1)\Phi_1^{(2)}(z_2)$$

$$-3\Phi_2^{(2)}(z_2)+4\Phi_1^{(1)}(z_1)\Phi_2^{(2)}(z_2)$$

Upon substitution we have

$$\hat{y}=5-2(z_1)+3(z_2)+5(z_1)(z_2)$$

$$-3(3z_2^2-2)+4(z_1)(3z_2^2-2)$$

$$=11-10z_1+3z_2+5z_1z_2-9z_2^2+12z_1z_2^2$$

Note that if the linear rule developed above is

applied pairwise to the vector of coefficients, followed by the application of the quadratic rule to the new vector, the resulting vector contains the coefficients of the reformulated expression.

Coefficients of Orthogonal Formulation	linear	quadratic
5	5	5-2(-3)=11
-2	3	-2-2(4)=-10
3	-3	3=3
5	-2	5=5
-3	5	3(-3)=-9
4	4	3(4)=12

Symbolically, the technique is as follows:

$$
\begin{aligned}
\hat{y} = {} & \hat{\beta}_0 + \hat{\beta}_1\Phi_1^{(1)}(z_1) + \hat{\beta}_2\Phi_1^{(2)}(z_2) \\
& + \hat{\beta}_3\Phi_1^{(1)}(z_1)\Phi_1^{(2)}(z_2) + \hat{\beta}_4\Phi_2^{(2)}(z_2) \\
& + \hat{\beta}_5\Phi_1^{(1)}(z_1)\Phi_2^{(2)}(z_2) \\
= {} & [\hat{\beta}_0 + \hat{\beta}_1\Phi_1^{(1)}(z_1)] + [\hat{\beta}_2 + \hat{\beta}_3\Phi_1^{(1)}(z_1)]\Phi_1^{(2)}(z_2) \\
& + [\hat{\beta}_4 + \hat{\beta}_5\Phi_1^{(1)}(z_1)]\Phi_2^{(2)}(z_2)
\end{aligned}
$$

The linear coefficients may be calculated first

$$
\begin{aligned}
= {} & [\hat{\beta}_0 + \hat{\beta}_1 z_1] + [\hat{\beta}_2 + \hat{\beta}_3 z_1]\Phi_1^{(2)}(z_2) \\
& + [\hat{\beta}_4 + \hat{\beta}_5 z_1]\Phi_2^{(2)}(z_2) \\
= {} & [\hat{\beta}_0 + \hat{\beta}_1 z_1] + [\hat{\beta}_2 + \hat{\beta}_3 z_1]z_2 + [\hat{\beta}_4 + \hat{\beta}_5 z_1][3z_2^2 - 2] \\
= {} & [\hat{\beta}_0 - 2\hat{\beta}_4] + [\hat{\beta}_1 - 2\hat{\beta}_5]z_1 + [\hat{\beta}_2 z_2] + [\hat{\beta}_3]\ z_1 z_2 \\
& + [3\hat{\beta}_4]\ z_2^2 + [3\hat{\beta}_5]\ z_1 z_2^2
\end{aligned}
$$

$\hat{\alpha}_0 = \hat{\beta}_0 - 2\hat{\beta}_4 \quad \hat{\alpha}_1 = \hat{\beta}_1 - 2\hat{\beta}_5 \quad \hat{\alpha}_2 = \hat{\beta}_2 \quad \hat{\alpha}_3 = \hat{\beta}_3$

$\hat{\alpha}_4 = 3\hat{\beta}_4 \quad \hat{\alpha}_5 = 3\hat{\beta}_5$

Rule: apply linear rule pair wise

apply quadratic rule in triplets

Example:		linear	quadratic
	$\hat{\beta}_0$	$\hat{\beta}_0$	$\hat{\beta}_0 - 2\hat{\beta}_4$
	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_1 - 2\hat{\beta}_5$
	$\hat{\beta}_2$	$\hat{\beta}_4$	$\hat{\beta}_2$
	$\hat{\beta}_3$	$\hat{\beta}_1$	$\hat{\beta}_3$
	$\hat{\beta}_4$	$\hat{\beta}_3$	$3\hat{\beta}_4$
	$\hat{\beta}_5$	$\hat{\beta}_5$	$3\hat{\beta}_5$

Multi-Factors

The general technique for the reformulation of the response function is

1. The ordering of the estimated coefficients of the orthogonal polynomials is to be maintained exactly as it resulted from the application of the Yates method.
2. One pass is made through the vector of coefficients for each factor in the design, i.e., a pass for the first factor, a pass for the second factor, etc.
3. Each pass consists of repeated applications of the linear, quadratic or cubic rule depending upon the number of levels for the corresponding

factor.

4. The resulting vector contains the coefficients of the reformulated polynomial.

The following is an example of the application of the "back solution" to the coefficients resulting from the application of the Yates method to a 3x2x4 factorial design. The last column contains the coefficients of the polynomial in terms of the standard z_i's.

	A Quadratic Rule	B Linear Rule	C Cubic Rule
$\hat{\beta}_0$	$\hat{\beta}_0 - 2\hat{\beta}_2$	$\hat{\beta}_0 - 2\hat{\beta}_2$	$[12(\hat{\beta}_0 - 2\hat{\beta}_2) - 15(\hat{\beta}_{12} - 2\hat{\beta}_{14})]/12$
$\hat{\beta}_1$	$\hat{\beta}_3 - 2\hat{\beta}_5$	$\hat{\beta}_6 - 2\hat{\beta}_8$	$[12\hat{\beta}_1 - 15\hat{\beta}_{13}]/12$
$\hat{\beta}_2$	$\hat{\beta}_6 - 2\hat{\beta}_8$	$\hat{\beta}_{12} - 2\hat{\beta}_{14}$	$[36\hat{\beta}_2 - 45\hat{\beta}_{14}]/12$
$\hat{\beta}_3$	$\hat{\beta}_9 - 2\hat{\beta}_{11}$	$\hat{\beta}_{18} - 2\hat{\beta}_{20}$	$[12(\hat{\beta}_3 - 2\hat{\beta}_5) - 15(\hat{\beta}_{15} - 2\hat{\beta}_{17})]/12$
$\hat{\beta}_4$	$\hat{\beta}_{12} - 2\hat{\beta}_{14}$	$\hat{\beta}_1$	$[12\hat{\beta}_4 - 15\hat{\beta}_{16}]/12$
$\hat{\beta}_5$	$\hat{\beta}_{15} - 2\hat{\beta}_{17}$	$\hat{\beta}_7$	$[36\hat{\beta}_5 - 45\hat{\beta}_{17}]/12$
$\hat{\beta}_6$	$\hat{\beta}_{18} - 2\hat{\beta}_{20}$	$\hat{\beta}_{13}$	$[36(\hat{\beta}_6 - 2\hat{\beta}_8) - 123(\hat{\beta}_{18} - 2\hat{\beta}_{20})]/12$
$\hat{\beta}_7$	$\hat{\beta}_{21} - 2\hat{\beta}_{23}$	$\hat{\beta}_{19}$	$[36\hat{\beta}_7 - 123\hat{\beta}_{19}]/12$
$\hat{\beta}_8$	$\hat{\beta}_1$	$3\hat{\beta}_2$	$[108\hat{\beta}_8 - 369\hat{\beta}_{20}]/12$
$\hat{\beta}_9$	$\hat{\beta}_4$	$3\hat{\beta}_8$	$[36(\hat{\beta}_9 - 2\hat{\beta}_{11}) - 123(\hat{\beta}_{21} - 2\hat{\beta}_{23})]/12$
$\hat{\beta}_{10}$	$\hat{\beta}_7$	$3\hat{\beta}_{14}$	$[36\hat{\beta}_{10} - 123\hat{\beta}_{22}]/12$
$\hat{\beta}_{11}$	$\hat{\beta}_{10}$	$3\hat{\beta}_{20} - 2\hat{\beta}_5$	$[108\hat{\beta}_{11} - 369\hat{\beta}_{23}]/12$
$\hat{\beta}_{12}$	$\hat{\beta}_{13}$	$\hat{\beta}_3 - 2\hat{\beta}_{11}$	$[27(\hat{\beta}_{12} - 2\hat{\beta}_{14})]/12$
$\hat{\beta}_{13}$	$\hat{\beta}_{16}$	$\hat{\beta}_9 - 2\hat{\beta}_{17}$	$[27\hat{\beta}_{13}]/12$
$\hat{\beta}_{14}$	$\hat{\beta}_{19}$	$\hat{\beta}_{15}$	$[81\hat{\beta}_{14}]/12$

	A Quadratic Rule	B Linear Rule	C Cubic Rule
$\hat\beta_{15}$	$\hat\beta_{22}$	$\hat\beta_{21}-2\hat\beta_{23}$	$[27(\hat\beta_{15} - 2\hat\beta_{17})]/12$
$\hat\beta_{16}$	$3\hat\beta_{2}$	$\hat\beta_{4}$	$[27\hat\beta_{16}]/12$
$\hat\beta_{17}$	$3\hat\beta_{5}$	$\hat\beta_{10}$	$[81\hat\beta_{17}]/12$
$\hat\beta_{18}$	$3\hat\beta_{8}$	$\hat\beta_{16}$	$[135(\hat\beta_{18} - 2\hat\beta_{20})]/12$
$\hat\beta_{19}$	$3\hat\beta_{11}$	$\hat\beta_{22}$	$[135\hat\beta_{19}]/12$
$\hat\beta_{20}$	$3\hat\beta_{14}$	$3\hat\beta_{5}$	$[405\hat\beta_{20}]/12$
$\hat\beta_{21}$	$3\hat\beta_{17}$	$3\hat\beta_{11}$	$[135(\hat\beta_{21}-2\hat\beta_{23})]/12$
$\hat\beta_{22}$	$3\hat\beta_{20}$	$3\hat\beta_{17}$	$[135\hat\beta_{22}]/12$
$\hat\beta_{23}$	$3\hat\beta_{23}$	$3\hat\beta_{23}$	$[405\hat\beta_{23}]/12$

2.1.5 Exchange of Yates Order

It will be seen in section 2.1.6 that the most efficient computational method for plotting the estimated response function against a specified factor over some fixed interval requires the factor of interest to have been specified last in the Yates order. Because the user of the program may wish to view several plots, each with respect to a different factor, the program must have the facility for rearranging the polynomial coefficients.

For example, assume that the program has produced for factors A(2 levels), B(3 levels), and C(2 levels) the following coefficients

Coefficient number	Factor levels A	B	C	Coefficients
1	1	1	1	100
2	2	1	1	- 10
3	1	2	1	20
4	2	2	1	30
5	1	3	1	- 40
6	2	3	1	- 20
7	1	1	2	10
8	2	1	2	0
9	1	2	2	50
10	2	2	2	- 30
11	1	3	2	40
12	2	3	2	- 50

If, however, the analysis had been performed with the factors in the order B, C, A, the results, although identical to the previous results, would have been arranged as follows

Coefficient number	Factor levels			Coefficients
	B	C	A	
1	1	1	1	100
2	2	1	1	20
3	3	1	1	- 40
4	1	2	1	10
5	2	2	1	50
6	3	2	1	40
7	1	1	2	- 10
8	2	1	2	30
9	3	1	2	- 20
10	1	2	2	0
11	2	2	2	- 30
12	3	2	2	- 50

If we wish to determine the location under the new factor ordering of one of the original coefficients, say 4, we first use the methods of section 2.1.2 to determine that the levels corresponding to position 4 are level 2(A), level 2 (B), and level 1(C). These levels may then be rearranged, i.e., level 2(B), level 1(C), and level 2(A), to determine, again by using the methods of section 2.1.2, that

$$\text{new location} = (2-1)66+(1-1)3+2$$
$$= 8$$

Thus, the coefficient 30, which was in position 4 under the original factor ordering, occupies position 8 under the re-

vised ordering.

Given an observation in location i, the corresponding factor settings are found by

$$j_1 = m + 1$$

$$j_k = [m/\prod_{\nu=1}^{k-1} \ell_\nu] + 1$$

where

$$m = (i-1)\bmod(\prod_{\nu=1}^{k} \ell_\nu)$$

and ℓ_ν is the level of the νth factor.

The j's and ℓ's may then be rearranged into the new ordering as, say, j_1', j_2', ..., j_n' and ℓ_1', ℓ_2', ... , ℓ_n' so that the new location i' is given by

$$i' = \sum_{k=2}^{n} (j_k'-1)\prod_{\nu=1}^{k-1} \ell_\nu' + j_1'$$

Example 2.1.5.1

Consider the example of section 2.1.2 in which it was found that for a 3x4x3x2 factorial design, the 60th observation occurs at settings 3, 4, 2 and 2, respectively. If the factors were to be rearranged into a 2x3x3x4 design as in the following table

new arrangement	old arrangement	ℓ'	j'
1	factor 4	2	2
2	factor 1	3	3
3	factor 3	3	2
4	factor 2	4	4

and, computationally,

$$\begin{aligned} i' &= (((j_4' - 1)\ell_3' + j_3 - 1)\ell_2' - 1)\ell_1' + j_1' \\ &= (((4 - 1)3 + 2 - 1)3 + 3 - 1)2 + 2 \\ &= 66 \end{aligned}$$

so that observation 60 under the original ordering of, say, ABCD becomes observation 66 of the new ordering DACB.

2.1.6 Mixed Polynomial Evaluation

The plotting of a response surface as a function of one of its factors requires the evaluation of a mixed polynomial expression. If the surface has been fitted using k factors and then if fixed values are substituted for k-1 of the variables while the kth variable is varied over an interval many such polynomial evaluations are required. Although it is possible to evaluate the expression say 100 times by simply substituting the values of the k variables at each of the 100 points, it is much more efficient to evaluate the expression in terms of the first k-1 variables only once and then to vary the kth variable over the interval. This reduction of a mixed polynomial expression to a simple single variable polynomial can be accomplished through the use of repeated synthetic divisions applied to polynomial coefficients arranged in the Yates order.

It is well known that the most efficient computational technique for the evaluation of a polynomial is that of synthetic division. In the case of a mixed poly-

nomial expression, the same technique can be used provided the coefficients are inversely arranged in the Yates order described in section 2.1.2.

Example 2.1.6.1

$$\begin{aligned} f(x,y) &= 10x^3y^2 - 14x^2y^2 + 8xy^2 - 12y^2 - 15x^3y \\ &\quad - 21x^2y + 12xy - 18y - 5x^3 + 7x^2 - 4x + 6 \\ &= (10x^3 - 14x^2 + 8x - 12)y^2 \\ &\quad + (15x^3 - 21x^2 + 12x - 18)y \\ &\quad + (-5x^3 + 7x^2 - 4x + 6) \\ &= ((10x^3 - 14x^2 + 8x - 12)y^2 \\ &\quad + (15x^3 - 21x^2 + 12x - 18)\ y \\ &\quad + (-5x^3 + 7x^2 - 4x + 6) \end{aligned}$$

Note that synthetic division can be applied to the 3 polynomials in x thus determining the coefficients of the y factors. Synthetic division may again be applied for the final evaluation of the expression.

Example 2.1.6.2

Consider the following mixed polynomial expression inversely arranged in the Yates order.

$$\begin{aligned} f(x,y,z) &= 20x^3y^2z - 28x^2y^2z + 16xy^2z - 24y^2z + 30x^3y \\ &\quad - 42x^2y^2 + 24xyz - 36yz - 10x^3z + 14x^2z - 8xz \\ &\quad + 12z + 10x^3y^2 - 14x^2y^2 + 8xy^2 - 12y^2 + 15x^3y \\ &\quad - 21x^2y + 12xy - 18y - 5x^3 + 7x^2 - 4x + 6 \end{aligned}$$

Suppose we wish to evaluate this expression for

$x = 1$, $y = 2$ and for $x = -1$. Figure 2.1.6.1 displays the step-by-step method of solution. The coefficients are listed in order on the far left. Because x is the first variable in the Yates ordering, synthetic division (for $x = 1$) is performed repeatedly to groups of 4 coefficients resulting in the 6 "collapsed" coefficients flagged with asterisks. These coefficients (of y,z terms) are copied into the middle column where synthetic division (for $y = 2$) is performed repeatedly to groups of 3 coefficients resulting in the 2 "collapsed" coefficients again flagged with asterisks. These coefficients (of z terms only) are copied into the last column where synthetic division (for $z=-1$) is performed once to produce the result (which is flagged with an asterisk). We see that

$$f(x=1,y=2,z=-1) = 48.$$

If, however, we wish to evaluate the expression for $x = 1$, $y = 2$ and for values of z ranging from say -10 to 10 in steps of 0.1, we see from the second column of figure 2.1.6.1 that

$$f(z|z=1,y=2) = -100z - 52$$

which can then be evaluated repeatedly for the desired values of z.

2.1.7 Confidence Intervals

The application of the Yates method to the analysis of data from a factorial experiment results in a simple

POLYNOMIAL EVALUATION BY REPEATED SYNTHETIC DIVISION

COEF OF (X,Y,Z)		X=1
20		20
-28	20	-8
16	-8	8
-24	8	-16*
30		30
-42	30	-12
24	-12	12
-36	12	-24*
-10		-10
14	-10	4
-8	4	-4
12	-4	8*
10		10
-14	10	-4
8	-4	4
-12	4	-8*
15		15
-21	15	-6
12	-6	6
-18	6	-12*
-5		-5
7	-5	2
-4	2	-2
6	-2	4*

COEF OF (Y,Z)		Y=2
-16		-16
-24	-32	-56
8	-108	-100*
-8		-8
-12	-16	-28
4	-56	-52*

COEF OF Z		Z=-1
-100		-100
-52	100	48*

FIGURE 2.1.6.1

computational procedure for the placement of confidence intervals about predicted responses. Chapter IV contains several examples of such confidence intervals. Examination of these displays naturally leads to the question of the location of the occurrence of the minimum and maximum distance of each interval from the predicted response. Direct application of the usual techniques provided by differential calculus supplies the answer.

Define q as follows:

$$q(z_o) = \frac{1}{n} + \frac{\Phi_o^2(z_o)}{\sum_z \Phi_o^2(z)} + \ldots + \frac{\Phi_{n-1}^2(z_o)}{\sum_z \Phi_{n-1}^2(z)}$$

where $\Phi_j(z)$ is the jth orthogonal polynomial for z at n levels and z_o is some fixed value for z.

Due to the orthogonality of Z'Z (Appendix A), we have in the one variable case

$$\begin{aligned} \mathrm{var}(\hat{y}|z_o) &= \sigma^2(1 + \underline{\Phi}'(z_o)(Z'Z)^{-1}\underline{\Phi}(z_o)) \\ &= \sigma^2(1 + q(z_o)) \end{aligned}$$

In the two variable case, say for z and w, we have

$$\mathrm{var}(\hat{y}|z_o,w_o) = \sigma^2(1 + q(z_o)q(w_o))$$

The extension to several variables is accomplished by simply introducing new q factors for the new variables.

Example 2.1.7.1

From the definition, in Appendix B, of the set of orthogonal polynomials for n=3, we see that for $z=z_o$ we have

$$q(z_o) = \frac{1}{3} + \frac{z_o^2}{2} + \frac{(3z_o^2-2)^2}{6}$$

It is of interest, then, to investigate the values which minimize and maximize q and, consequently, the confidence interval distances about y.

a. 2 levels

$$q = \frac{1}{2} + \frac{z^2}{2}$$

attains a maximum (q = 1) and $z = \pm 1$ (domain limits) and a minimum ($q = \frac{1}{2}$) at z = 0.

b. 3 levels

$$q = \frac{1}{3} + \frac{z^2}{2} + \frac{(3z^2-2)^2}{6}$$

attains a maximum (q = 1) at z = 0 and $z = \pm 1$ and a minimum ($q = \frac{5}{8}$) at $z = \pm 1/\sqrt{2}$

c. 4 levels

$$q = \frac{1}{4} + \frac{(3z)^2}{20} + \frac{(\frac{9}{4}z^2-\frac{5}{4})^2}{4} + \frac{(\frac{45}{4}z^3-\frac{41}{4}z)^2}{20}$$

attains a maximum ($q \doteq 1.178$) at $z = \pm\sqrt{\frac{73-2\sqrt{301}}{135}}$ and a minimum (q = 41/64) at z = 0. (Note that at factor settings $z = \pm 1$ and $z = \pm 1/3$, q = 1 and a local minimum ($q \doteq 0.748$) occurs at $z = \pm\sqrt{\frac{73+2\sqrt{301}}{135}}$.)

In the case of several factors we see that, again due to orthogonality, if we wish to calculate confidence

intervals for a range of one factor z with all other factors fixed, we simply calculate the q values once for the fixed factors and then evaluate q(z) for each z value of interest. Note that the maxima and minima occur at the values of z found above even in the presence of other fixed factors.

2.1.8 Calculation of Contours

The plotting of response surface contours first requires the evaluation of a mixed polynomial expression for k-2 of its k factor variables. Secondly, the expected response for which contours are desired is subtracted from the constant term of the polynomial, a value for the (k-1)st variable is substituted into the reduced expression and, finally, the roots of the remaining single variable polynomial are extracted producing a point on the contour.

The methodology for accomplishing this is as follows: The two variables for which contours are desired are selected and the coefficients of the expression are rearranged, in the Yates order, using the method of section 2.1.5, so that of the two variables of interest, the one with the smaller number of levels is placed last and the other one next to last. The remaining factors precede these two in any order. By the use of the method of section 2.1.6 the expression is reduced to one involving only the last two variables. The desired response value is then subtracted

from the constant term. One of the selected variables (i.e., the next to last factor in the Yates order) is then fixed at each of 51 points, chosen for effective spacing on the display screen, covering the desired interval; thus the expression is reduced to a polynomial in one variable. The roots of this polynomial are calculated and those which lie between the requested limits for the variable which occurs last in the Yates order constitute the specified contour.

Example 2.1.8.1

For the three variable polynomial in section 2.1.6, set x = 1 and find those values of $1 \le y \le 3$ and $-2 \le z \le 0$, if any, for which

$$f(x,y,z) = 48$$

From section 2.1.6, the reduced polynomial is

$$f(y,z|x=1) = -16y^2z - 24yz + 8z - 8y^2 - 12y + 4$$

Now, find y and z such that

$$f(y,z|z=1) = 48$$

or

$$-16y^2z - 24yz + 8z - 8y^2 - 12y - 44 = 0$$

Using 3 values of y rather than the 50 values used in practice, the scheme illustrated in Figure 2.1.8.1 is produced.

CALCULATION OF CONTOURS

For y = 1, then, as in section 2.1.6, the reduction is as follows

-16		-16
-24	-16	-40
8	-40	-32*
-8		-8
-12	-8	-20
-44	-20	-64*

thus $-32z - 64 = 0$

$z = -2$

For y = 2, then

-16		-16
-24	-32	-56
8	-108	-100*
-8		-8
-12	-16	-28
-44	-56	-100*

thus $-100z - 100 = 0$

$z = -1$

For y = 3, then

-16		-16
-24	-48	-72
8	-216	-208*
-8		-8
-12	-24	-36
-44	-108	-152

thus $-208z - 152 = 0$

$z = -19/26$

FIGURE 2.1.8.1

So that for $x = 1$ and $f(y,z|x=1) = 48$, three points along the contour are

y	z
1	-2
2	-1
3	-19/26

2.2 Polynomial Approximation Using the Cubic Spline

Given a set of points (x_1,y_1), (x_2,y_2), ... , (x_n,y_n) arranged in increasing values of x, we wish to find a function which can be used to interpolate for given intermediate values of x. Using the technique presented in Scott and Norman[23], the solution can be accomplished through the use of a cubic spline function, i.e.,

$$y = f(x) = a + bx + \sum_{i=1}^{n} c_i (x - x_i)_+^3$$

$$\text{where} \quad (d)_+^3 = d^3 \qquad \text{for } d > 0$$
$$= 0 \qquad \text{otherwise}$$

In order to determine the c_i's one employs the following technique

x	y	f[,]	g
x_1	y_1		
		f_1	
x_2	y_2		g_2
		f_2	
.	.		g_3
.	.	.	
.	.	.	.
		.	.
			.
			g_{n-1}
		f_{n-1}	
x_n	y_n		

where

$$f_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}$$

and

$$g_i = f_i - f_i - 1$$

The values of $s_2, s_3, \ldots, s_{n-1}$ may then be determined by solving the following tri-diagonal system.

$$\begin{bmatrix} 2(x_3-x_1) & (x_3-x_2) & 0 & \ldots & 0 & 0 \\ 0 & 2(x_4-x_2) & (x_4-x_3) & & & 0 \\ \vdots & & & & & \vdots \\ 0 & & & & 2(x_{n-1}-x_{n-3}) & (x_{n-1}-x_{n-2}) \\ 0 & 0 & & \ldots & & 2(x_n - x_{n-2}) \end{bmatrix} \begin{bmatrix} s_2 \\ s_3 \\ \vdots \\ s_{n-2} \\ s_{n-1} \end{bmatrix} = \begin{bmatrix} g_2 \\ g_3 \\ \vdots \\ g_{n-2} \\ g_{n-1} \end{bmatrix}$$

Upon solving this system, the c_i's may be found by

x	s	h[,]	c
	0		
		0	
x_1	0		c_1
		$s_2/(x_2-x_1)$	
x_2	s_2		c_2
		$(s_3-s_2)/(x_3-x_2)$	
x_3	s_3		.
		.	.
.	.	.	.
.	.	.	
.	.		
x_{n-2}	s_{n-2}		
		$(s_{n-1}-s_{n-2})/(x_{n-1}-x_{n-2})$	c_{n-2}
x_{n-1}	s_{n-1}		
		$-s_{n-1}/(x_n-x_{n-1})$	c_{n-1}
x_n	0	0	c_n
	0		

where $$c_i = h_{i+1} - h_i$$

The linear coefficients, a and b, may then be found by direct evaluation at x_1 and x_2.

CHAPTER III

COMPUTATIONAL IMPLEMENTATION

The mathematical procedures described in the previous chapter may be transcribed into a set of subprograms which can then be combined to produce interactive graphics programs. Two such programs, as described in Chapter I, have been produced. Sections 3.1 and 3.2 of this chapter provide documentation for each of the subroutines of the mathematical procedures used by the system. In addition, complete instructions are given in section 3.3 for the preparation of data for disk input to the YATES program. For the statistican who wishes to add his own function subprograms to the Graphics Distribution Unit, section 3.4 contains the steps necessary for the permanent or temporary inclusion of these routines into the library. Certain utility subroutines necessary for the implementation of the programs are found in section 3.5.

3.1 The YATES Program

The description of the main program includes a flow chart of the program logic. Following this description is a detailed list of all computational subroutines used.

3.1.1 The Main Program

a. Loading instructions

$LINK YATES

b. Description

The YATES program is an interactive graphics program prepared for the analysis of data resulting from a complete factorial design. The experiment to be analyzed may include up to 4 factors, with each factor having up to 4 levels. A maximum of 6 replications of the experiment may be input to the program. The major capabilities of the program are: (1) the computation of the analysis of variance table, (2) the calculation of the response equation, (3) the plotting of cross-sections of the response surface (with or without confidence intervals), and (4) the plotting of contours of the response surface.

c. Subroutines called

PLOTS, BEGIN, INPUT, EDIT, YATES, BACK, PAGIT, PLOTTER

d. COMMON areas

YATDAT, DIRAC

e. Mode of Operation

computational

f. Logical Flow

The flow of the main program is displayed in Figure 3.1.1.

FLOWCHART OF YATES - PART I

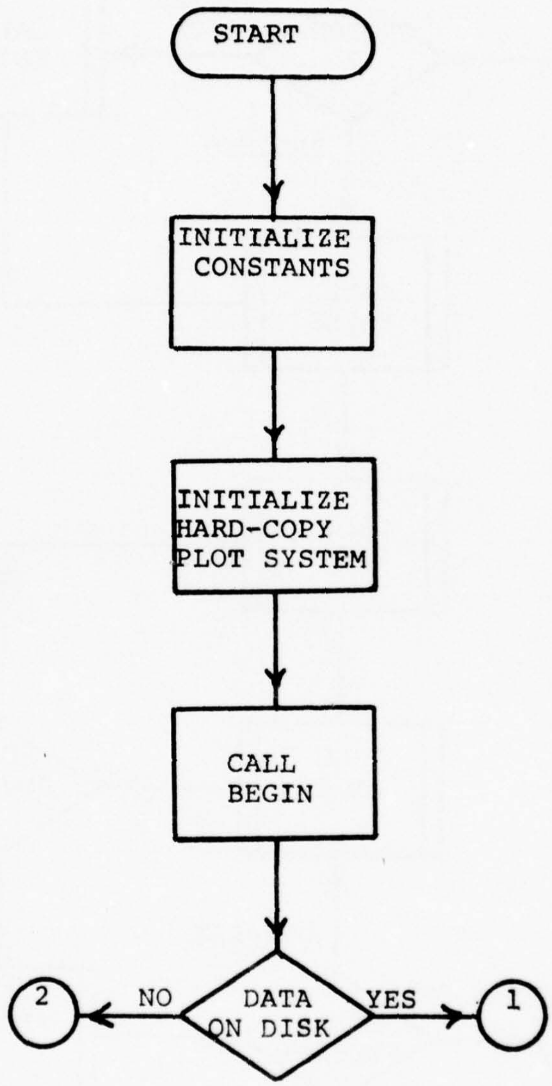


FIGURE 3.1.1a

FLOWCHART OF YATES - PART 2

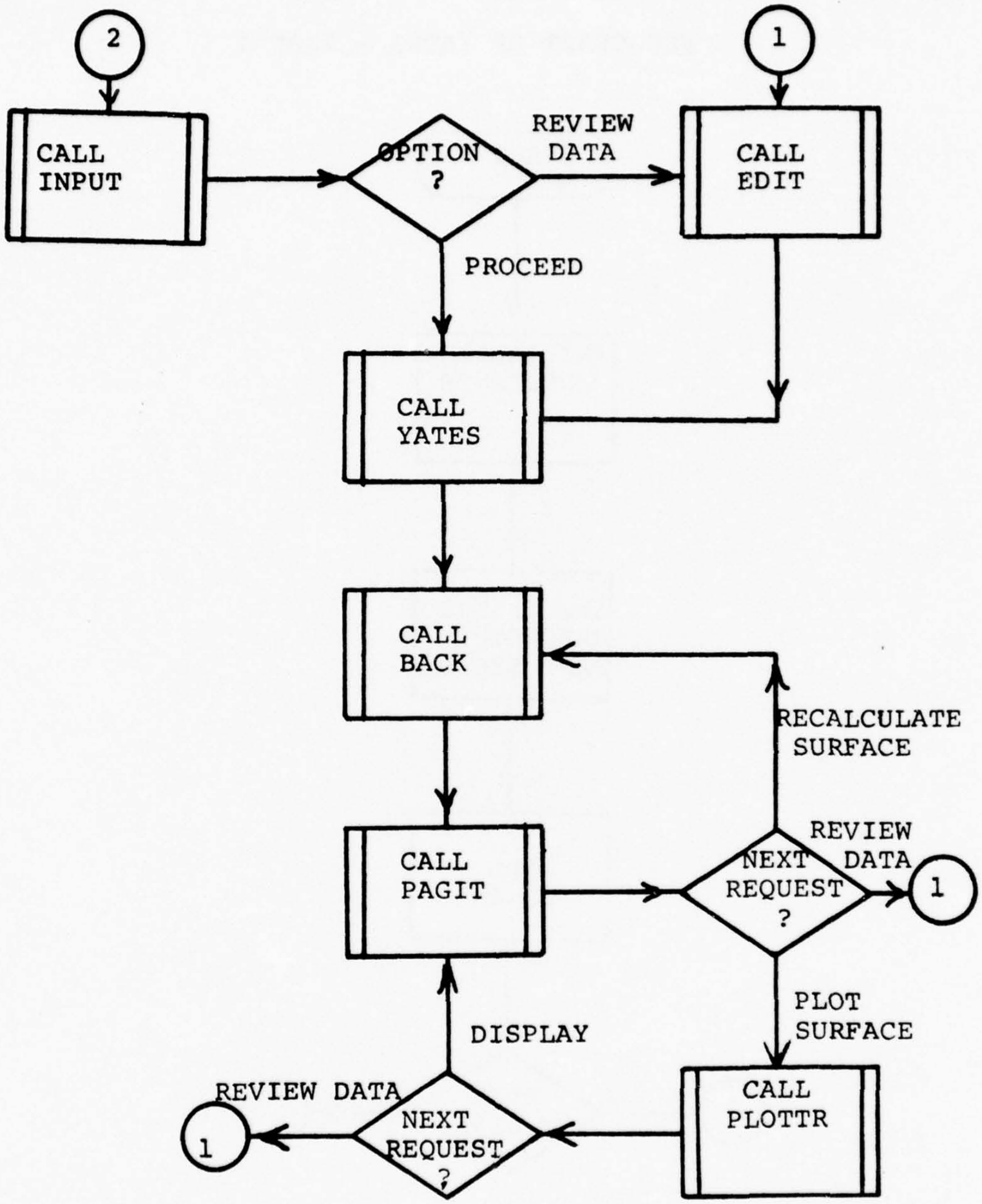


FIGURE 3.1.1b

3.1.2 Subroutine BEGIN

a. Calling Sequence

CALL BEGIN(&NUM1, &NUM2)

&NUM1 - The statement number to which control is returned if user wishes to enter data from the keyboard.

&NUM2 - The statement number to which control is returned if input data are stored on disk.

b. Description

This subroutine displays the initial frame to the user. Program limitations are listed and the user is provided with the option to enter his data from the keyboard or to read it directly into the program from disk.

c. Subroutines called

GRINIT, EORINT, GCEOR, MASKIT, GRDPLY, SPACE, TWAIT, GERAS, GRRLSE

d. COMMON Areas

unlabeled COMMON, DIRAC

e. Mode of Operation

COMFORT

Subroutine INPUT

a. Calling Sequence

CALL INPUT(&NUM1, &NUM2)

&NUM1 - The statement number to which control is returned if the user wishes to edit the raw data.

&NUM2 - The statement number to which control is returned if the user wishes to continue with the analysis of his data.

b. Description

This subroutine requests from the user the observations to be analyzed, along with the additional information necessary for the analysis: the number of factors, the number of levels for each factor, the number of replications, the factor settings and, additionally, the mean square error if only one replication is present in the data. The data input is prompted in Yates order and the user is given the opportunity to edit a line of input at any time. Upon completion of the input the data are stored away on disk and the user is provided with the option to either review his data or to continue with the analysis. If the user chooses to continue with the analysis, he is given the option to retain or

immediately delete all terms whose corresponding single degrees of freedom F values are less than 1.0.

c. Subroutines called

GRINIT, EOBINT, GCEOB, MASKIT, GRDPLY, TWAIT, GRRPLX, INX, GERAS, FETCH, SPACE, ANAL, GBKSP, GRRLSE

d. COMMON areas

unlabeled COMMON, BAKDAT, YAIDAT, DIRAC

e. Mode of Operation

COMFORT

f. Logical Flow

The logical flow of the INPUT subroutine is as follows

i. The INPUT subroutine determines interactively the number of factors in the design, the number of replications, the name of the factors, and the number of levels for each factor. If the data consist of only one replication, the user is asked to supply an estimate of the Mean Square Error and the number of degrees of freedom on which the estimate is based.

ii. The subroutine then determines, from the user, the values for the level settings of the factors. A default of level 1 = 1, level 2 = 2, etc.

is provided.

iii. The subroutine then generates and displays, one at a time and in the Yates order, all of the combinations of level settings for the factors. The user responds by entering, from the keyboard, all of the replications for each setting. The data are stored away on disk.

iv. The user may interrupt the process in step (iii) at any time in order to edit any previously entered data.

v. After completion of data entry, all of the information gathered in steps (i) and (ii) is stored away on disk.

vi. Finally, the user is presented with the option of deleting each response equation coefficient whose corresponding single degree of freedom F value is less than 1.

3.1.4 Subroutine EDIT

a. Calling Sequence

CALL EDIT

b. Description

This subroutine reads the raw data from disk and displays it to the user. The user may either simply view the data or replace observations with new data. Upon completion of the editing of the data, the user is given the option to retain or immediately delete all terms whose corresponding single degree of freedom F values are less than 1.0. The analysis of the data follows.

c. Subroutine called

GRINIT, EOBINT, GCEOB, GRDPLY, MASKIT, TWAIT, GERAS, SPACE, GRRPLX, INX, FETCH, ANAL, GBKSP, GRRLSE

d. COMMON areas

unlabeled COMMON, BAKDAT, YATDAT, DIRAC

e. Mode of Operation

COMFORT

3.1.5 Subroutine YATES

a. Calling Sequence

CALL YATES

b. Description

This subroutine uses the Yates Method to perform an analysis of variance for a factorial design and to calculate the coefficients of the orthogonal polynomials used to approximate the response surface. The analysis of variance results are formatted to disk for later presentation to the user. If the user has requested deletion of terms whose corresponding single degree of freedom F values are less than 1.0, this is done and the sum of squares for these terms is added into the error sum of squares. Note: the analysis of variance is not modified.

c. Subroutine called

ANAL

d. COMMON areas

DIRAC, YATDAT, BAKDAT

e. Mode of Operation

Computational

f. Logical Flow

The logical flow of the YATES subroutine is as follows

i. The subroutine first examines the number of

replications, NREP. If NREP equals 1, an estimate, requested by the INPUT subroutine, of the mean square error, VSE, for the response variable and the number of degrees of freedom, NER, on which these estimates are based are used. If NREP is greater than 1, an estimate of the mean square error for the response variable is calculated from the data by subtracting from the total sum of squares, the sum of squares of the cell totals divided by the number of replications and dividing that result by its associated number of degrees of freedom, the number of replication minus one times the total number of cells.

ii. The subroutine then calculates $d_1 \times d_2$ (as illustrated in Appendix C. An element of the array d_1, called ND1 in the subroutine, is defined to be the number of replications times the number times the particular treatment combination present in that cell, ignoring all factors at level 1, appears as a factor in other cells. Computationally, each factor setting is scanned using ANAL. ND1 is calculated as the number of replications times the product of the total number of design levels for each factor with a level of 1 in that particular cell. An element of the array d_2, called ND in the subroutine, is the

product of the sums of squares of the orthogonal polynomials corresponding to each level setting of the factors in a particular cell. Because d_1 and d_2 are not used separately in the subroutine, the product $d_1 \times d_2$ is saved in the array ND1.

iii. The Yates algorithm is then employed to calculate the necessary linear combinations of the cell totals, DAT. One pass is made through the cell total vector for each level of each factor in the design with the resulting vector for each factor replacing the cell total vector before moving to the next factor. The linear combinations used are found in section 2.1.3. After having performed the above calculations for each factor, with the final results stored in DAT, the estimates, array A, of the coefficients of the response equation are calculated by dividing each element of the array DAT by the corresponding element of the array ND1. The single degree of freedom sums of squares, array SS, are similarly calculated by dividing the square of each element of the array DAT by corresponding element of the array ND1. The coefficients and sums of squares are then written out to logical unit NW.

iv. In order to provide an analysis of variance table

reporting the main effects and all interactions, the single degree of freedom sums of square are then accumulated into the vector SQA which is of order $2^{NFAC}-1$. Simultaneously, the vector NR (also of order $2^{NFAC}-1$) is used to accumulate the corresponding degrees of freedom. A binary placement technique is used to find the proper position in SQA for a particular single degree of freedom sum of squares, so that, for example, an analysis of variance table for factors A, B and C will have the main effect of C in position 4, the AB, AC, BC and ABC interactions in positions 3, 5, 6 and 7, respectively. After the sums of squares have been accumulated, the subroutine ANAL is used, with all levels equal to 2, to generate the factor names for the analysis of variance table. As the line counter, KH for the table passes from 1 through $2^{NFAC}-1$, the levels generated by ANAL are 1 for the absence of a factor and 2 for the presence of a factor. This information is used to load the output vector SETUP with the proper factor names which are then written to disk in print format -- for later display on the IBM 2250 screen -- along with the corresponding degrees of freedom, from NR, the sums of squares divided by the degrees of freedom) and the F value (the mean square divided by the mean square error, VSE).

After the analysis of variance table has been completed for all main effects and interactions, a line identified as <u>all effects</u> is also written to disk. This

line contains the degrees of freedom for all effects, NALL (the number of cells minus one), the sum of squares for all effects, SQAL (the accumulated sums of squares for all main effects and interactions) and the corresponding mean square for all main effects and interactions) and the corresponding mean square and F value. Lastly, a line identified as error is written. This line contains the degrees of freedom for error, the sum of squares for error and the mean square error as calculated or requested by INPUT.

3.1.6 Subroutine BACK

a. Calling Sequence

CALL BACK

b. Description

This subroutine converts the coefficients of the orthogonal polynomial representation of the response surface into coefficients in terms of the standardized z values. It then formats this form of the equation to disk for later presentation to the user. In addition, it calculates the equations necessary to transform the z values into the original raw values. These operations are also formatted to disk.

c. Subroutines called

ANAL, DWMULT

d. COMMON areas

YATDAT, BAKDAT, DIRAC

e. Mode of Operation

Computational

f. Logical Flow

The logical flow of the BACK subroutine is as follows

i. The vector DAT, which was passed from the YATES subroutine, contains the coefficients of the orthogonal polynomial form of the response equation.

The vector DAT is copied into a vector A. Linear combinations, as specified in section 2.1.4, of the vector A are made -- one complete pass for each factor. The resulting vector contains the coefficients of the polynomial in terms of the original standard values.

ii. The coefficients for the transformations necessary to convert the raw values for each factor into the standard values are calculated using the methods specified in section 2.1.2.

iii. Using the mean square error VSE from subroutine YATES, the minimum and maximum two sigma confidence limits are calculated as in section 2.1.7.

iv. The actual response equation in standard values is the written to disk in print format for later display on the IBM 2250 screen.

v. Also placed on disk in a similar format are the raw value to standard value transformations for the factors.

vi. Similarly, the minimum and maximum two sigma confidence limits are formatted to disk for later display.

3.1.7 Subroutine PAGIT

a. Calling Sequence

CALL PAGIT (&NUM1, &NUM2, &NUM3)

&NUM1 - The statement number to which con- is passed if the user wishes to edit the raw data.

&NUM2 - The statement number to which control is passed if the user wishes to plot the response surface.

&NUM3 - The statement number to which control is passed if the user wishes to recalculate the response equation after deleting one or more orthogonal terms.

b. Description

This subroutine simply displays from disk the analysis of variance table, the response equation and, for the statistician, the single degree of freedom sums of squares. The user may (1) return to the raw data for editing, (2) plot the response surface or (3) delete one or more orthogonal coefficients and recalculate the standardized equation of the response surface (the Analysis of Variance table is unchanged by this process).

c. Subroutines called

EOBINT, GRINIT, GCEOB, NEWPGE, MASKIT, STUFF, SNAP, TWAIT, GERAS, SPACE, GRDPLY, FETCH, ANAL, GRRPLX, INX, GRRLSE

d. COMMON areas

BAKDAT, YATDAT, DIRAC, unlabeled COMMON

e. Mode of Operation

COMFORT

3.1.8 Subroutine PLOTTR

a. Calling Sequence

CALL PLOTTR (&NUM1, &NUM2)

&NUM1 - The statement number to which control is passed if the user wishes to edit the raw data.

&NUM2 - The statement number to which control is passed if the user wishes to view the analysis of variance table and response equation.

b. Description

This subroutine allows the user to choose to plot (1) cross-sections of the response surface, (2) confidence intervals about predicted values or (3) contours of the response surface. The function of this routine is one of control.

c. Subroutine called

EOBINT, GRINIT, GCEOB, MASKIT, NEWPGE, STUFF, SNAP, TWAIT, GERAS, GRRLSE, CONTUR, CONF, SPLOT

d. COMMON areas

unlabeled COMMON

e. Mode of Operation

COMFORT

3.19 Subroutine SPLOT

a. Calling Sequence

CALL SPLOT

b. Description

This subroutine plots the response surface against a selected factor with all other factors set at some specified set of values. Up to four plots may be viewed simultaneously. If the user attempts to extrapolate, a warning is given.

c. Subroutines called

EORINT, INITP, PASKIT, EOBP, PEWPGE, GRAXES, PSTUFF, PSNAP, PWAIT, GREPLX, INX, PPLUCK, PEWLINE, FETCH, REFILL, GRCHAR, UVOO, XCHNGE, BNAL, POLY, SPOLY, BPOLY, PUTUV, GRGRID, GRPLOT, UV99, RLSEP

d. COMMON areas

unlabeled COMMON, YATDAT

e. Mode of Operation

COMPLOT

f. Logical Flow

The logical flow of the SPLOT subroutine is as follows:

i. The subroutine first determines, from the user, the factor to be used as the abscissa.

ii. It then determines, again from the user, the minimum and maximum values to be used for the abscissa. If the user requests values which will result in extrapolation, he is so warned.

iii. The values to be used for the remaining factors are obtained from the user. Multiple values may be entered for each factor so long as the total number of plots does not exceed 4.

iv. The coefficients of the polynomial are rearranged from the original Yates order into a new Yates order for which the plotting factor is considered the last factor. The method used is explained in section 2.1.5.

v. As in section 2.1.6, the reordered polynomial is collapsed, using the first set of specified factor values, into one in terms of the plotting factor only.

vi. For 51 points, equally spaced from the user-indicated minimum to the maximum, the subroutine uses the reduced polynomial from step v. to calculate and plot the predicted response.

vii. Steps v. and vi. are repeated until the requested number of plots have been displayed.

3.1.10 Subroutine CONF

a. Calling Sequence

CALL CONF

b. Description

The subroutine plots the response surface displayed with upper and lower 95% confidence limits against a selected factor with all other factors set a some specified set of values. Only one plot may be viewed per frame. If the user attempts to extrapolate, a warning is given.

c. Subroutines called

EOBINT, INITP, EOBP, PEWPGE, GRAXES, PASKIT, PSTUFF, PSNAP, PWAIT, GREPLX, INX, P LUCK, PEWLINE, FETCH, REFILL, GRCHAR, UVOO, EXCHNGE, POLY, ORTH, ANAL, SPOLY, BPOLY, PUTUV, GRPLOT, UV99, RLSEP

d. COMMON areas

unlabeled COMMON, YATDAT, BAKDAT, DIRAC

e. Mode of Operation

COMPLOT

f. Logical Flow

The logical flow of the CONF subroutine is as follows

i. The subroutine first determines, from the user, the factor to be used as the abscissa.

ii. It then determines, again from the user, the minimum and maximum values to be used for the abscissa. If the user requests values which will result in extrapolation, he is so warned.

iii. The (single) values to be used for the remaining factors are obtained from the user.

iv. The coefficients of the polynomial are rearranged from the original Yates order into a new Yates order for which the plotting factor is considered the last factor. The method used is explained in section 2.1.5.

v. The q values (for the confidence intervals) defined in section 2.1.7 are calculated for all the factors, omitting the q value for the plotting factor.

vi. As in section 2.1.6, the reordered polynomial is collapsed, using the (single) specified factor values, into one in terms of the plotting factor only.

vii. For 51 points equally spaced from the user-indicated minimum to the maximum, the subroutine uses the reduced polynomial from step vi. to calculate and plot the predicted response.

viii. Additionally, the remaining q values are calculated for each of the 51 points and the corresponding confidence intervals are calculated and plotted -- the character L is used for the lower limit; U, for the upper limit.

3.1.11 Subroutine CONTUR

a. Calling Sequence

CALL CONTUR

b. Description

This subroutine plots contours of the response surface for given values of the predicted response against two selected factors with all other factors set at some specified set of values. Up to four contours may be viewed simultaneously. If the user attempts to extrapolate, a warning is given.

c. Subroutines called

EOBINT, INITP, EOBP, PEWPGE, GRAXES, PASKIT, PSTUFF, PSNAP, PWAIT, GREPLX, INX, ZTORAW, PPLUCK, DEWLINE, FETCH, REFILL, GRCGAR, UVOO, ANAL, LTNEW, POLY, BPOLY, ROOTER, PUTUV, GRGRID, GRPLOT, UV99, RLSEP

d. COMMON areas

unlabeled COMMON, YATDAT, DIRAC

e. Mode of Operation

COMPLOT

f. Logical Flow

The logical flow of the CONTUR subroutine is as follows

i. The subroutine first determines, from the user, the two factors which are to be plotted against

each other.

ii. It then determines, again from the user, the minimum and maximum values for which each of the factors is to be plotted.

iii. The (single) values to be used for the remaining factors (if any) are determined from the user.

iv. The predicted response for which the contour is to be plotted is then read in from the user.

v. The coefficients of the polynomial are rearranged from the original Yates order into a new Yates order for which the plotting factor with the smaller number of levels has the last position; and the other plotting factor, next to last position.

vi. As in section 2.1.6, the polynomial is then collapsed, using the (single) specified factor values, into one in terms of the two plotting factors only.

vii. The predicted response is then subtracted from the constant term of the polynomial.

viii. For 51 points, equally spaced from minimum to maximum, the polynomial (now in two variables, adjusted

for the predicted value) is further reduced to one involving only the last factor. The roots of this resulting polynomial are extracted. Those roots within the user-specified range are paired with the corresponding next-to-last factor value and are subsequently plotted. If no acceptable roots are found for any of the 51 points, a message of this condition is displayed to the user.

ix. The subroutine allows the user the option of typing in another predicted response, causing steps vii through viii to be repeated. This procedure continues until a fifth plot is requested. At that time the four previous plots are erased from the screen and the fifth plot is placed on the screen as the first of a new sequence of four plots.

3.1.13 Subroutine ANAL

a. Calling Sequence

CALL ANAL(NN,I1,I2,I3,I4)

NN - An integer which indicates the Yates order position

Ik - An integer which contains the Yates setting for the kth factor.

b. Description

This function receives as input the position to which a vector of factor settings would belong for a given Yates ordering. It returns, as in section 2.1.2, the vector of factor settings.

c. Subroutines called

none

d. COMMON areas

YATDAT

e. Mode of Operation

computational

3.1.13 Subroutine BNAL

a. Calling Sequence

CALL BNAL(NN,I1,I2,I3,I4,LVL)

NN - Refer to ANAL

Ik - Refer to ANAL

LVL - An integer vector which contains the levels of the factors.

b. Description

Refer to ANAL

c. Subroutines called

none

d. COMMON areas

none

e. Mode of Operation

computational

3.1.14 Function Subprogram LT

a. Calling Sequence

NY = LT(I1,I2,I3,I4)

Ik - An integer which contains the YATES setting for the kth factor.

b. Description

This function receives as input the factor settings for up to 4 factors and returns the position to which this vector would belong in the YATES order. The method is presented in section 2.1.2.

c. Subroutines called

none

d. COMMON areas

YATDAT

e. Mode of Operation

computational

3.1.15 Function Subprogram LTNEW

a. Calling Sequence

NY = LTNEW(LVL,I1,I2,I3,I4)

LVL - An integer vector which contains the levels of the factors.

Ik - Refer to LT.

b. Description

Refer to LT.

c. Subroutines called

none

d. COMMON areas

none

e. Mode of Operation

computational

3.1.17 Subroutines POLY and BPOLY

a. Calling Sequence

CALL POLY(A,X,N)

CALL BPOLY(A,X,N)

A - A real vector containing the coefficients (in decreasing order for POLY; increasing for BPOLY)

N - The degree of the polynomial

X - The value at which to evaluate the polynomial

b. Description

This subroutine uses synthetic division, as described in section 2.1.6, to evaluate a simple polynomial expression.

c. Subroutines called

none

d. COMMON areas

none

e. Mode of Operation

computational

3.1.18 Subroutine ROOTER

a. Calling Sequence

CALL ROOTER(S,R,NR,LVL)

S - A real vector which contains the coefficients (in increasing order) of the polynomial whose roots are to be found.

R - A real vector which will contain the roots of the polynomial

NR - The number of roots extracted.

LVL - The number of coefficients in S

b. Description

This subroutine extracts the roots of a polynomial of degree 3 or less.

c. Subroutines called

none

d. COMMON areas

none

e. Mode of Operation

computational

3.1.19 Subroutine SPOLY

a. Calling Sequence

CALL SPOLY(B,VAL,ISTOP,NORDR,A)

- B - A real vector containing coefficients of a mixed polynomial arranged in the YATES order.
- VAL - A real vector of values for the variables for which the polynomial is to be collapsed.
- ISTOP - The number of variables for which the polynomial is to be collapsed.
- NORDER- The order of the variables relative to the original order,
- A - A real vector containing the rearranged coefficients.

b. Description

This subroutine collapses a mixed polynomial expression arranged in the Yates order by the technique specified in section 2.1.6.

c. Subroutines called

BPOLY

d. COMMON areas

YATDAT

e. Mode of Operation

computational

3.1.20 Subroutine XCHNGE

a. Calling Sequence

CALL XCHNGE(JF,COF)

JF - An integer which indicates a factor number with which to interchange the last factor.

COF - A vector which will return the coefficients arranged in the revised YATES order.

b. Description

This subroutine interchanges the coefficients of the response surface coefficients under the assumption that factor JF and the last factor to be input have been exchanged. The procedure followed can be found in section 2.1.5.

c. Subroutines called

ANAL, LTNEW

d. COMMON areas

YATDAT

e. Mode of Operation

computational

3.1.21 Subroutine ZTORAW

a. Calling Sequence

CALL ZTORAW(C,J)

C - A real vector which returns the coefficients of the polynomial required to convert a z value back to its corresponding raw value.

J - An integer which indicates the factor for which the transformation coefficients are to be calculated.

b. Description

This subroutine provides the coefficients of the polynomial which is used to convert a z value back to the corresponding raw value. The process is accomplished directly from the coefficients of the inverse transformation which are located in the YATDAT COMMON area.

c. Subroutines called

none

d. COMMON areas

YATDAT

e. Mode of Operation

computational

3.2 The BLOWUP Program

This section describes the computer implementation of the spline technique for interpolation of statistical functions. The results of the interpolation may be displayed in plot or table form.

3.2.1 The Main Program

a. Loading instructions

$LINK BLOWUP

b. Description

This program uses spline interpolation to display a range of values for any function available in the GMS calculator mode. Actual calculations are made for 11 equally spaced points. A spline fit is then made to these points. The user may employ this spline function to plot the resulting curve or to display tables of the values.

c. Subroutines called

EOBINT, PFINT, PLOTS, GRINIT, GCEOB, NEWPGE, GCPFK, STUFF, SNAP, TWAIT, GRRPLX, INX, FETCH, GRRLSE, INITP, PFKP, EOBP, GRAXES, GRCHAR, GRGRID, UVOO, ENSYM, EVALL, PUTUV, SPLCON, GRPLOT, UV99, SPLINE, PWAIT, REFILL, RLSEP, GREPLY, XBLANK

d. COMMON areas

none

e. Mode of Operation

COMFORT, COMPLOT

f. Logical Flow

The flow of the program is displayed in Figures 3.2.1 and 3.2.2.

FLOWCHART OF INPUT TO BLOWUP

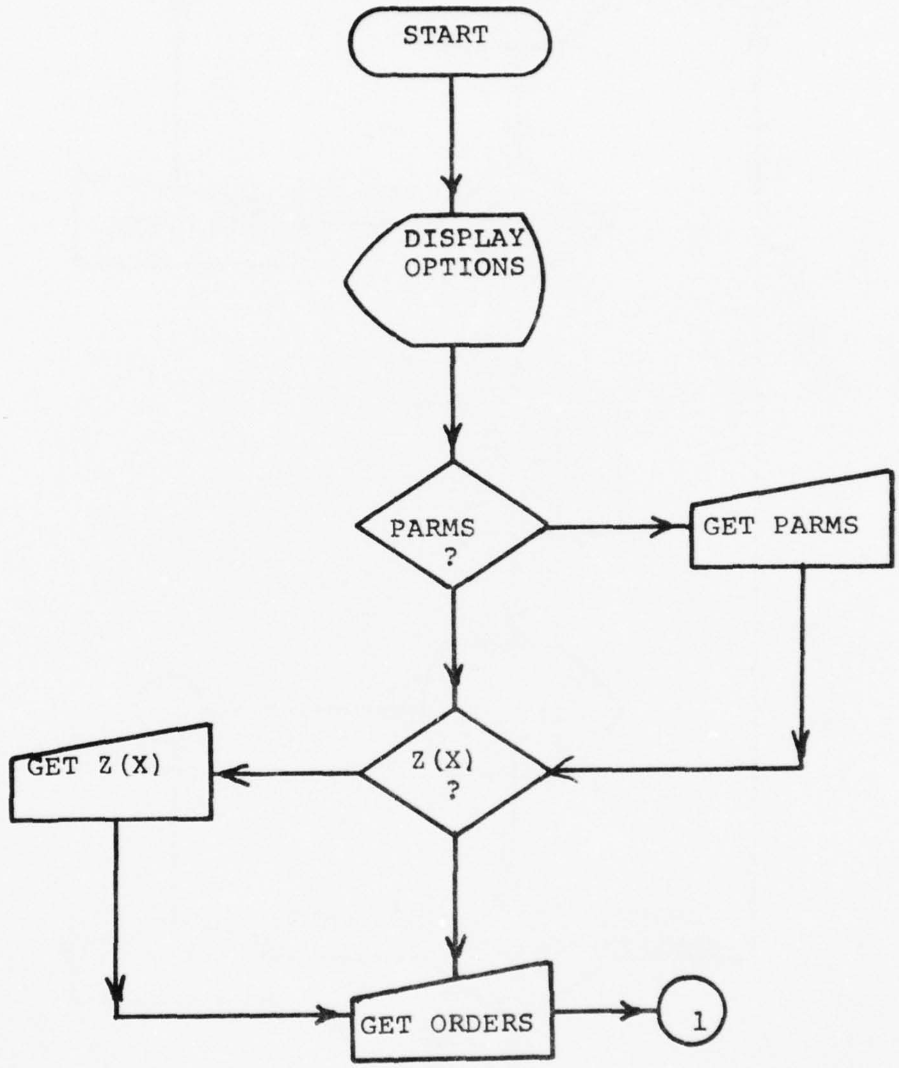


FIGURE 3.2.1

FLOWCHART OF BLOWUP PLOT ROUTINE

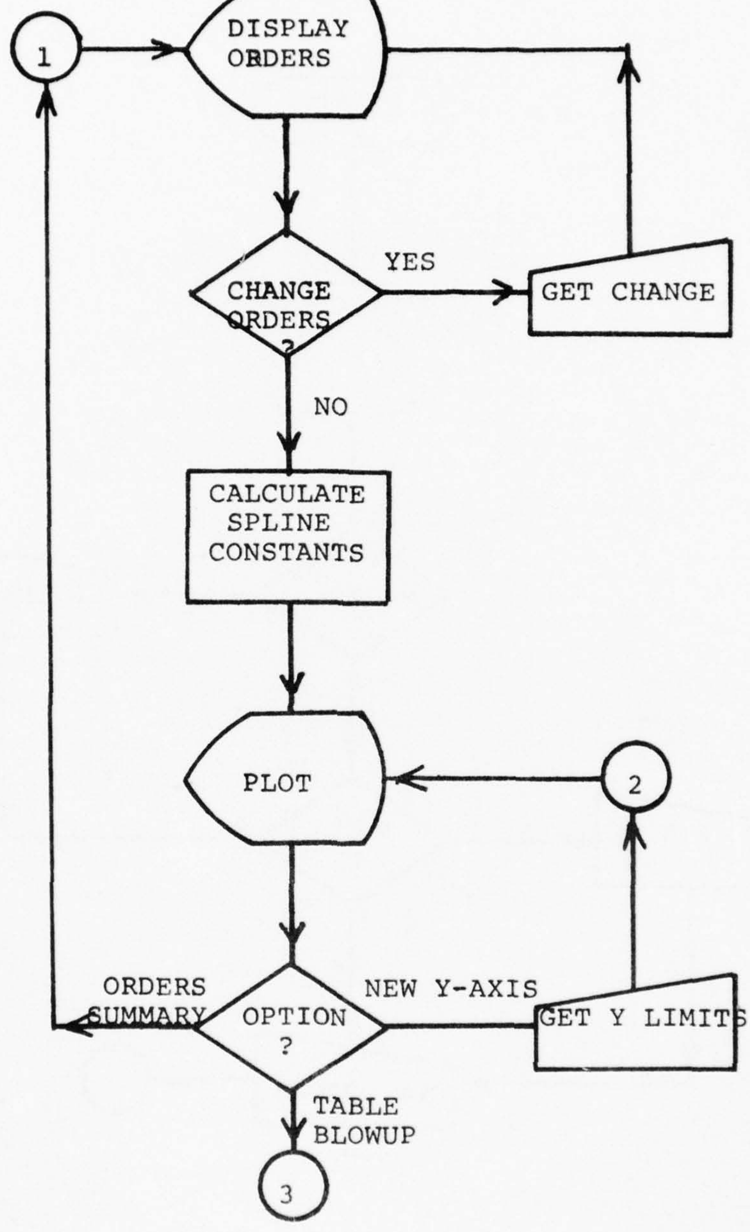


FIGURE 3.2.2a

FLOWCHART OF BLOWUP PLOT ROUTINE

(CONTINUED)

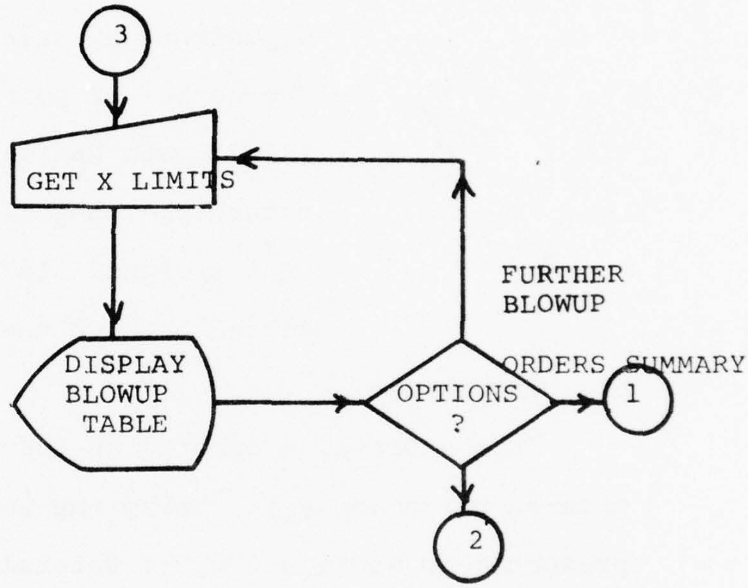


FIGURE 3.2.2b

3.2.2 Subroutine SPLCON

a. Calling Sequence

CALL SPLCON(X,Y,M,C)

- X - A real vector containing the independent variable.
- Y - A real vector containing the dependent variable.
- M - The number of points to fit.
- C - A 4x11 real matrix which will return the coefficients of the cubic polynomials which have been fitted through the X,Y pairs.

b. Description

This subroutine accepts as input a set of ordered pairs (x_i,y_i). Using the spline technique presented in section 2.2, it determines the coefficients of the cubic polynomials which pass between the points.

c. Subroutines called

none

d. COMMON areas

none

d. Mode of Operation

computational

3.2.3 Subroutine SPLINE

a. Calling Sequence

CALL SPLINE(X,Y,M,C,XINT,YINT)

X - A real vector containing the independent variable.

Y - A real vector containing the depencent variable.

M - The number of points.

C - A 4x11 real matrix which contains the coefficients of the previously fitted cubics.

XINT - The X value for which the interpolation is to be made.

YINT - The returned interpolated value of Y.

b. Description

This subroutine uses the coefficients determined by the SPLCON subroutine to return an interpolated value of Y for a given value of X.

c. Subroutines called

none

d. COMMON areas

none

e. Mode of Operation

computational

3.3 Disk Input to YATES

Even for a factorial experiment of moderate size it becomes a time-consuming task to type in the data to the YATES program. Additionally, the user who may wish to examine the response surface for his data during several terminal sessions would, justifiably, reject the prospect of re-typing his data on each occasion. For these reasons, the system has been written so that a user may "spool" his data to a permanent disk for later input to the program. It is important for the potential user of this program to know the proper format for the storage of the data.

The assumption is that the data are to be processed by a FORTRAN program. The use of any other programming language would simply require the translation of these specifications into those compatible with the other language. The file is to be set up as a FORTRAN direct-access unformatted file with the following FORTRAN statement

```
DEFINE FILE 19(270,6,U,IREC)
```

which indicates that

a. the program will write to unit 19

b. the file is to consist of a maximum of 270 records

c. the record length is fixed at 6 words (24 bytes)

d. the file is to be created with unformatted WRITEs

e. the record pointer is the integer variable IREC

The following variables are to be DIMENSIONed exactly as shown

```
DIMENSION LVL(4),NAME(8),CONCOM(4,4),REP(6)
```

where

LVL(I) is the number of levels for factor I.

NAME(2*I-1) and NAME(2*I) contain the (up to) 8 character name of factor I.

CONCOM(I,J) is the setting for the J*th* level of factor I.

The following information must be written as shown

```
WRITE(19'257) NF,NREP,LVL
WRITE(19'260) NAME,CONCOM,VSE
```

where

NF is the number of factors.

NREP is the number of replications

VSE is an estimate of the mean square error (required only if NREP = 1)

LVL, NAME and CONCOM are as defined above.

The observations are to be written sequentially in the Yates order (see Chapter 2) with factor 1 varying most rapidly, factor 2 next, etc. Each record contains all replications (limit of 6) for each factor combination.

Example:

```
      DO 50 I = 1,48
         . . .
      WRITE(19'I) REP
   50 CONTINUE
```

where the vector REP contains all of the observations made at each factor combination. Note that because of the predetermined ordering, the actual values of the factor settings do not have to be written into each record. This information is determined from CONCOM.

3.4 The Addition of a Function to the BLOWUP Program

Any double precision FORTRAN function subprogram, the arguments of which are all in double precision floating point, may be added to the BLOWUP program.

3.4.1 Permanent Additions

The procedure for permanent addition consists of adding the name of the function to a table in INTERP [17], compiling the FORTRAN function and, finally, link-editing the BLOWUP program. To aid in the explanation of the process, suppose that we wish to add the double precision function, say, FUN to the system. The first step is to add the function name to the FUNCNMS table in the assembler program INTERP. The entry should be made after the last existing entry and before the statement

```
#FUNC   EQU   (*-FUNCNMS)/8
```

The new entry should be punched as follows:

```
DC   CL8'FUN'
```

After this step has been done, the process may be completed as in Figure 3.4.1.

3.4.2 Temporary Additions

The function names FUN1, FUN2, FUN3 and FUN4 have been reserved for temporary use. In order to add a function to the library temporarily, the user may give his function

one of these reserved names.

JOB CONTROL LANGUAGE

```
//         (standard JOB card)
//*MAIN   SYSTEM=SY65
//STEP1 EXEC ASMFC
//ASM.SYSLIB DD DSN=SYS1.UGAMAC,DISP=SHR,DCB=BLKSIZE=3520
// DD DSN=SYS1.MACLIB,DISP=SHR
//ASM.SYSIN DD *
      (The INTERP source with the table modification)
//STEP2 EXEC FORTGCL
//FORT.SYSIN DD *
      (The FORTRAN function to be added)
//LKED.SYSLMOD DD DSN=SYS1.GRAPHLIB(BLOWUP),DISP=SHR,
//   SPACE=(TRK,(1,0,1))
//LKED.TEMP DD DSN=SYS1.GRAPHLIB,DISP=SHR
//LKED.SYSIN DD *
 INCLUDE TEMP(BLOWUP)
 ENTRY MAIN
/*
```

Figure 3.4.1

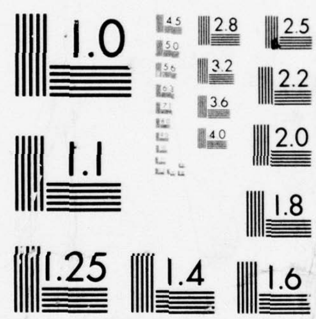
Example:

```
FUNCTION FUN3(A,B,C)
```

Only STEP2 of the JCL listing of Figure 3.4.1 is to be used in this case. STEP1 should be completely omitted. The added function will remain in the system until replaced by a different function with the name FUN3. At present, FUN1 evaluates the c.d.f. of a largest characteristic root.

AD-A036 346 GEORGIA UNIV ATHENS DEPT OF STATISTICS AND COMPUTER--ETC F/G 9/2
GRAPHICAL AIDS FOR STATISTICAL COMPUTATION.(U)
DEC 76 W P BOND, R E BARGMANN N00014-69-A-0423
UNCLASSIFIED TR-112 NL

2 OF 4
AD
A036346



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

3.5 Utility Subroutines

The implementation of the interactive graphics programs described in this chapter required the creation of a number of non-mathematical systems subprograms. The routines were written to facilitate the processing of both input from the IBM 2250 graphics console, and output the display screen. The purpose of this section is to describe the operaion of these routines.

3.5.1 Subroutine CHECK

a. Calling Sequence

CALL CHECK(&NUM1,&NUM2,MODE)

&NUM1 - The statement number to which control is returned if an unexpected response is made by the user.

&NUM2 - The statement number to which control is returned if the user indicates termination of the program.

MODE - This variable must have been set 1 if the program is in COMFORT mo or to 2 if the program is in COMP mode.

b. Description

The subroutine first determines whether or not

the user has depressed key 31; if so, return is made to the second statement number in the calling sequence. If not, the subroutine determines whether or not the user has depressed key 15; if so, all of the lights on the program function keyboard are turned on, SCOPLT is called to generate hard-copy plot instructions to disk and the program function keyboard lights are restored to their original state. Return is then made to the first statement number in the calling sequence. If the type of interrupt is not one that is expected of ir a key which is not enabled is depressed, a return is made to the first statement number in the calling sequence; otherwise, a standard return is made.

c. Subroutines called

PFINT, GCPFK, PFKP, SCOPLT

c. COMMON areas

unlabeled COMMON

d. Mode of Operation

COMFORT, COMPLOT

3.5.2 Subroutine GREPLX

a. Calling Sequence

CALL GREPLX (TEXT, NC)

Refer to GRRPLX for argument definition.

b. Description

Refer to GRRPLX

c. Subroutines called

GREPLY, XBLANK

d. COMMON areas

none

e. Mode of Operation

COMPLOT

3.5.3 Subroutine GRRPLX

a. Calling Sequence

CALL GRRPLX (TEXT, NC)

TEXT - The location of the area into which to place the text string to be received from the keyboard.

NC - The maximum number of characters of text expected.

b. Description

The subroutine immediately issues a wait for a reply from the keyboard. When the reply is received, XBLANK is called to remove all blanks from the text string. A standard return is then made.

c. Subroutine called

GRRPLY, XBLANK

d. COMMON areas

none

e. Mode of Operation

COMFORT

3.5.4 Subroutine MASKIT

a. Calling Sequence

CALL MASKIT (K)

K - an index to the desired program function key mask

b. Description

The subroutine contains a table of all program functions by masks which will be used during the execution of the program. Upon entry, the K mask is selected and placed in unlabeled COMMON and program function key lights corresponding to 1 bits in the mask are lighted.

c. Subroutine called

GCPFK

d. COMMON areas

unlabeled COMMON

e. Mode of Operation

COMFORT

3.5.5 Subroutine NEWPGE (STUFF, SNAP, PLUCK, NEWLINE)

a. Calling Sequences (five entry points)

CALL NEWPGE

CALL STUFF (K, ALINE)

CALL SNAP

CALL PLUCK (K, ALINE)

CALL NEWLINE (K, ALINE)

K - An integer specifying an appropriate number.

ALINE - The location of a 68 character text string.

b. Description

An entry at NEWPGE causes an output buffer, OUT, of 3240 (or 45 x 72) characters to be set to blanks with every 72nd character set to hexadecimal '15' (EOL code).

An entry at STUFF causes the k^{th} line of the output buffer to be replaced with the contents of ALINE.

An entry at SNAP causes the output buffer to be displayed to the user.

An entry at PLUCK causes the K^{th} line of the output buffer to be copied to ALINE for return to the calling routine.

An entry at NEWLNE causes the K^{th} line of the

output buffer to be set to blanks.

c. Subroutine called

GERAS, GRDPLY

d. COMMON areas

none

e. Mode of Operation

COMFORT

3.5.6 Subroutine PASKIT

a. Calling Sequence

CALL PASKIT (K)

(refer to MASKIT for argument definition)

b. Description

refer to MASKIT

c. Subroutine called

PFKP

d. COMMON areas

unlabeled COMMON

e. Mode of Operation

COMPLOT

3.5.7 Subroutine PEWPGE (PSTUFF, PSNAP, PPLUCK, PEWLNE)

A. Calling Sequences (five entry points)

CALL PEWPGE

CALL PSTUFF (K, ALINE)

CALL PSNAP

CALL PPLUCK (K, ALINE)

CALL PEWLINE (K, ALINE)

(refer to NEWPGE for argument definition)

b. Description

(refer to NEWPGE)

c. Subroutine called

REFILL, GRCHAR

d. COMMON areas

none

e. Mode of Operation

COMPLOT

3.5.8 Subroutine PWAIT

A. Calling Sequence

CALL PWAIT (I, J, K, L)

(refer to TWAIT for argument definition)

b. Description

Refer to TWAIT

c. Subroutines called

GWAIT, CHECK, PLOT, RLSEP

d. COMMON areas

unlabeled COMMON

e. Mode of Operation

COMPLOT

3.5.9 Subroutine SPACE

a. Calling Sequence

CALL SPACE (NS)

NS - the number of blank lines to be displayed to the user

b. Description

The subroutine causes NS blank lines to be displayed to the user.

c. Subroutine called

GRDPLY

d. COMMON areas

none

e. Mode of Operation

COMFORT

3.5.10 Subroutine TWAIT

a. Calling Sequence

CALL TWAIT (I, J, K, L)

I - 1 if program function by interrupt is to be allowed; 0, otherwise.

J - 1 if light pen interrupt is to be allowed; 0, otherwise.

K - 1 if keyboard interrupt is to be allowed; 0, otherwise.

L - 1 if time interrupt is to be allowed; 0, otherwise.

b. Description

This subroutine places the program in a wait-state by calling the subroutine GWAIT. Immediately after the occurrence of an interrupt, the subroutine CHECK is called. A nonstandard RETURN 1 causes the subroutine to close the graphics system, to call PLOT in order to close the hardcopy file, and then to terminate execution of the program. A nonstandard RETURN2 returns to the original call to GWAIT; thus, ignoring the interrupt. A standard return from CHECK causes TWAIT to issue a standard return.

c. Subroutine called

GWAIT, CHECK, PLOT, GRRLSE

d. COMMON areas

unlabeled COMMON

e. Mode of Operation

COMFORT

CHAPTER IV

APPLICATIONS OF RESPONSE SURFACE UNIT

Many experimental situations involve the examination of the quantity or quality of the output of a process which has several factors as input. Normally the first analysis performed on data gathered from this type of experiment is an analysis of variance. This statistical technique allows the experimenter to examine, not only the significance of the effects of the factors, but also the effects of the interactions of several factors. In many cases, the experimenter is attempting to find a specific combination of factor settings which will maximize the yield of the process. This is usually accomplished by fitting mixed polynomials through the response measured at each setting of the factors. This polynomial approximation may then be used to interpolate, so as to locate the maximum value. The ideal situation would be to fit the surface only to responses in the vicinity of the maximum and, to study the characteristics of the surface by graphing cross-sections or projection of the surface for each of the several variables. However, attempts to sketch, by hand, the various 2-dimensional projections of a response surface

are so time-consuming that most users would prefer to resort to numerical approaches in studying the surface characteristics near the suspected maximum. Many numerical methods have been developed to locate the maximum of a function of several variables. It should be pointed out that these methods were developed to circumvent the difficulty of visualizing the surface and thus do not involve characteristics that could be obtained from graphical representations. However, all of these numerical methods require some initial values for the input variables. If the surface is involved, even in the best of numerical methods, an improper choice of initial values may lead to solutions far away from the optimum. The interactive graphics programs presented here have been developed to enable the experimenter to use the computer in an effective fashion to visualize many different projections of an intricate surface.

4.1 Response Surfaces

To illustrate the use of the interactive YATES program for the analysis of data from a factorial experiment, this section follows a step-by-step interactive session during which the user enters the data to the program, performs an analysis of variance and finally, calculates and plots, in various formats, the response surface. This example is based on data appearing in

Davies [7], page 519, Table 11.4. The illustration involves a 3^2 factorial experiment designed so as to facilitate the location of the point of maximum yield. Figures 4.1.1 through 4.1.30 represent actual displays as they appear on the screen of an IBM 2250 graphics terminal. Figure 4.1.1 presents the capabilities and limitations of the program to the user. Note that data may be entered from the keyboard, as in this example, or they may have been stored previously on a disk through a program operating in batch mode, as described in section 3.3.

In figure 4.1.2 the user has indicated to the program that he has two factors and only one replication per cell. As it is not possible to estimate the mean square error with only one observation per cell, the program asks the user to supply an estimate. As suggested by Davies [7], prior knowledge of the process provides an estimate of 0.6, which has been entered. In Figure 4.1.3 the program is asking for the names by which the user wishes to refer to his factors. For convenience, the user may refer to them as FACA and FACB; however, in this case the user enters the names X1 and X2 as in Davies [7].

In figure 4.1.4 the user has replied that factor X1 has three levels, as has factor X2. At this point the user may correct any input errors that he might have made. In Figure 4.1.5 the user has been given the opportunity to default to values of 1, 2, 3, etc. or to enter the actual

DAVIES EXAMPLE - FRAME 1

OUTPUT AREA

THIS UNIT WILL PERFORM AN ANALYSIS OF VARIANCE FOR ANY COMPLETE FACTORIAL DESIGN OF THE FORM

$$2^R 3^S 4^T$$

WITH R+S+T LESS THAN OR EQUAL TO 4

FOR UP TO 6 REPLICATIONS.

IF DESIRED, THE UNIT WILL ALSO DETERMINE THE EQUATION OF THE RESPONSE SURFACE AND, FURTHER, WILL ALLOW THE USER TO EXAMINE PLOTS OF THIS SURFACE.

AT ANY POINT DURING THE SESSION, THE USER MAY RETURN TO THE GMS MONITOR BY DEPRESSING PROGRAM FUNCTION KEY 31.

PLEASE DEPRESS EITHER
KEY 1 IF YOU WISH TO ENTER DATA, OR
KEY 2 IF YOUR DATA ARE STORED ON DISK.

REPLY AREA

FIGURE 4.1.1

DAVIES EXAMPLE - FRAME 2

OUTPUT AREA

```
PLEASE ENTER THE NUMBER OF FACTORS.
2
PLEASE ENTER THE NUMBER OF REPLICATIONS.
1
ENTER AN ESTIMATE OF THE MSE.
```

REPLY AREA

FIGURE 4.1.2

DAVIES EXAMPLE - FRAME 3

OUTPUT AREA

```
DEPRESS KEY 1 TO USE DEFAULT NAMES FACA, FACB, ETC.
OR ENTER THE NAME OF THE FIRST FACTOR (LIMIT 8 LETTERS).
X1
PLEASE ENTER THE NAME OF THE LAST FACTOR.
```

REPLY AREA

FIGURE 4.1.3

DAVIES EXAMPLE - FRAME 4

OUTPUT AREA

THERE WILL BE AN OPPORTUNITY AT THE END OF THIS FRAME
TO CORRECT ANY INPUT ERRORS.
PLEASE ENTER THE NUMBER OF LEVELS FOR X1
FACTOR 1 HAS 3 LEVELS.

PLEASE ENTER THE NUMBER OF LEVELS FOR X2
FACTOR 2 HAS 3 LEVELS.

DEPRESS KEY 1 TO CONTINUE OR KEY 2 TO RE-ENTER LEVEL

REPLY AREA

FIGURE 4.1.4

DAVIES EXAMPLE - FRAME 5

OUTPUT AREA

IF YOU WISH THE LEVELS OF THE FACTORS TO BE
1 FOR LEVEL 1, 2 FOR LEVEL 2, ETC. DEPRESS KEY 1

IF YOU WISH TO ENTER VALUES FOR THE LEVELS, DEPRESS KEY 2

REPLY AREA

FIGURE 4.1.5

settings for later response surface analysis. The settings do not have to be equally spaced. In Figure 4.1.6 the user has entered the values -1, 0 and 1 for both factors as suggested by Davies [7].

In Figure 4.1.7 the program is asking for input of the raw data. A has been identified as X1 and B, as X2. In Figure 4.1.8 five of the observations have been entered, 71.7 for levels 1 of X1 and X2, 79.2 for level 2 of X1 and level 1 of X2, etc. In Figure 4.1.9 the data entry is complete. The user has the option of editing the data or performing the analysis.

In Figure 4.1.10, the user has elected to proceed with the analysis and, consequently, one final question has been asked. The user may delete all coefficients whose corresponding F's are less than one or he may choose to view all of the coefficients calculated. In this case the user depresses key 2.

Figure 4.1.11 displays the first frame of the analysis of variance, the sums of squares for the main effects and interactions. Note that the error term was entered *a priori* and has one degree of freedom associated with it. Figure 4.1.12, the second frame of the analysis of variance, displays the mean squares for the effects listed above. Figure 4.1.13 displays the F values based on the mean square error of 0.6 which was entered externally.

Figure 4.1.14 displays the full response equation as

DAVIES EXAMPLE - FRAME 6

OUTPUT AREA

IF YOU WISH THE LEVELS OF THE FACTORS TO BE
1 FOR LEVEL 1, 2 FOR LEVEL 2, ETC. DEPRESS KEY 1

IF YOU WISH TO ENTER VALUES FOR THE LEVELS, DEPRESS KEY 2

ENTER, SEPARATED BY COMMAS, THE LEVELS FOR X1

-1,0,1

ENTER, SEPARATED BY COMMAS, THE LEVELS FOR X2

-1,0,1

DEPRESS KEY 1 TO CONTINUE

REPLY AREA

FIGURE 4.1.6

DAVIES EXAMPLE - FRAME 7

OUTPUT AREA

ENTER, SEPARATED BY COMMAS, ALL REPLICATIONS FOR THE FACTOR LEVELS INDICATED

A = X1   B = X2

DEPRESS KEY 30 TO EDIT A LINE OF DATA.

CELL	A	B
1	1	1

REPLY AREA

FIGURE 4.1.7

DAVIES EXAMPLE - FRAME 8

OUTPUT AREA

ENTER, SEPARATED BY COMMAS, ALL REPLICATIONS FOR THE FACTOR LEVELS INDICATED

A = X1 B = X2

DEPRESS KEY 30 TO EDIT A LINE OF DATA.

CELL	A	B	
1	1	1	71.7
2	2	1	79.2
3	3	1	80.1
4	1	2	75.2
5	2	2	81.5
6	3	2	

REPLY AREA

FIGURE 4.1.8

DAVIES EXAMPLE - FRAME 9

OUTPUT AREA

A = X1 B = X2

DEPRESS KEY 30 TO EDIT A LINE OF DATA.

CELL	A	B	
1	1	1	71.7
2	2	1	79.2
3	3	1	80.1
4	1	2	75.2
5	2	2	81.5
6	3	2	79.1
7	1	3	76.3
8	2	3	80.2
9	3	3	75.8

DATA ENTRY IS COMPLETE.
DEPRESS KEY 1 TO PERFORM ANALYSIS
DEPRESS KEY 2 TO REVIEW OR EDIT DATA.

REPLY AREA

FIGURE 4.1.9

DAVIES EXAMPLE - FRAME 10

OUTPUT AREA

DEPRESS KEY 1 TO DELETE EACH SINGLE DEGREE OF FREEDOM F LESS THAN 1.0
DEPRESS KEY 2 TO RETAIN ALL F VALUES

REPLY AREA

FIGURE 4.1.10

DAVIES EXAMPLE - FRAME 11

OUTPUT AREA

ANALYSIS OF VARIANCE

SOURCE OF VARIATION			DF	SUM OF SQUARES
X1	*		2	54.1489
X2	*		2	4.10893
X1	*X2	*	4	20.2177
ALL EFFECTS			8	78.4755
ERROR			1	0.600000

DEPRESS KEY 1 FOR SUMS OF SQUARES.
DEPRESS KEY 2 FOR MEAN SQUARES.
DEPRESS KEY 3 FOR F VALJES.

DEPRESS KEY 30 TO VIEW OPTION TABLE

REPLY AREA

FIGURE 4.1.11

DAVIES EXAMPLE - FRAME 12

OUTPUT AREA

ANALYSIS OF VARIANCE

SOURCE OF VARIATION			DF	MEAN SQUARES
X1	*		2	27.0744
X2	*		2	2.05446
X1	*X2	*	4	5.05443
ALL EFFECTS			8	9.80944
ERROR			1	0.600000

DEPRESS KEY 1 FOR SUMS OF SQUARES.
DEPRESS KEY 2 FOR MEAN SQUARES.
DEPRESS KEY 3 FOR F VALUES.

DEPRESS KEY 30 TO 'IEW OPTION TABLE

REPLY AREA

FIGURE 4.1.12

DAVIES EXAMPLE - FRAME 13

OUTPUT AREA

ANALYSIS OF VARIANCE

SOURCE OF VARIATION	DF	F
X1 *	2	45.12
X2 *	2	3.424
X1 *X2 *	4	8.424
ALL EFFECTS	8	16.35
ERROR	1	

DEPRESS KEY 1 FOR SUMS OF SQUARES.
DEPRESS KEY 2 FOR MEAN SQUARES.
DEPRESS KEY 3 FOR F VALUES.

DEPRESS KEY 30 TO VIEW OPTION TABLE

REPLY AREA

FIGURE 4.1.13

DAVIES EXAMPLE - FRAME 14

```
                       OUTPUT AREA
             FACTOR*A  IS  X1
             FACTOR*B  IS  X2

       IN STANDARD FACTOR VALUES, Y =
+  81.49995        +  1.949996    *A
 -4.350003     *A**2    + 0.4999999     *B
 -2.224998     *A*B   -0.4250030    *A**2 *B
 -1.800008     *B**2    + 0.2500152E-01*A*B**2
+ 0.6249998     *A**2 *B**2

   WHERE*A    = 0.0            * ARAW**2   + (   1.000000      )* ARAW
+ ( 0.0          )
   WHERE*B    = 0.0            * BRAW**2   + (   1.000000      )* BRAW
+ ( 0.0          )
   MINIMUM TWO SIGMA ERROR IS + OR -         1.63299
   MAXIMUM TWO SIGMA ERROR IS + OR -         2.25092
```

```
DEPRESS KEY 28 TO PAGE BACKWARD.
DEPRESS KEY 29 TO PAGE FORWARD.

DEPRESS KEY 30 TO VIEW OPTION TABLE
```

REPLY AREA

FIGURE 4.1.14

calculated. Note that since all of the coefficients have been left in the equation regardless of the F value, the equation does not agree with that in Davies [7] which contains only the coefficients of X1, X2, $X1^2$, $X2^2$, X1X2 and the intercept. Figure 4.1.15 has coefficients of the orthogonal polynomials and the corresponding single degree of freedom sums of squares and F ratios, to enable the user to eliminate single degrees of freedom which, in his opinion, are not relevant. In this session, coefficients are deleted when the F values are less than one. Their sums of squares are then added the error. This eliminates effects B linear (4), A quadratic B linear (6), A linear B quadratic (8) and A quadratic B quadratic (9). Figure 4.1.16 displays the reduced equation. The only remaining terms are those in X1, $X1^2$, X1X2 and $X2^2$. Note that Davies [7] kept the X2 term even though, as he points out, its coefficient cannot be shown to be significantly different from zero.

Figure 4.1.17 displays the options available to the user. The user chooses to plot the response surface against one factor and therefore depresses key 1. In Figure 4.1.18 the user requests a plot of the response surface against X1. In Figure 4..19 the user states that X1 is to run from -1 to +1 and that there will be a plot for each of three values of X2, -1, 0 and +1. Figure 4.1.20 is the requested plot. Note that of these plots, plot 2 (X2 = 0) achieves the largest predicted yield of 81.5 for an X1 value of about 0.3.

DAVIES EXAMPLE - FRAME 15

OUTPUT AREA

SINGLE DEGREE OF FREEDOM DISPLAY

DEPRESS KEY 1 TO DELETE TERMS BY CELL NUMBER
DEPRESS KEY 2 TO DELETE TERMS BY F COMPARISON
DEPRESS KEY 28 TO PAGE BACKWARD
DEPRESS KEY 29 TO PAGE FORWARD

DEPRESS KEY 30 TO VIEW OPTION TABLE

MSE = 0.59999996

A = X1 B = X2

CELL	A	B	COEFFICIENT	SUM OF SQUARES	F RATIO
1	0	0	77.677734	54304.469	90507.438
2	1	0	1.9666643	23.206604	38.677673
3	2	0	-1.3111124	30.942307	51.570511
4	0	1	0.21666461	0.28166133	0.46943557
5	1	1	-2.2249985	19.802400	33.004089
6	2	1	-0.14166766	0.24083674	0.40139455
7	0	2	-0.46111381	3.8272667	6.3787775
8	1	2	0.83338395E-02	0.83343498E-03	0.13890583E-02
9	2	2	0.69444418E-01	0.17361110	0.28935182

REPLY AREA

FIGURE 4.1.15

DAVIES EXAMPLE - FRAME 16

```
                         OUTPUT AREA
               FACTOR*A   IS  X1
               FACTOR*B   IS  X2

         IN STANDARD FACTOR VALUES, Y =
+  81.22218       +  1.966664     *A
 -3.933337     *A**2     -2.224998     *A*B
 -1.383341     *B**2

   WHERE*A    = 0.0             * ARAW**2  + (   1.000000     )* ARAW
+ ( 0.0           )
   WHERE*B    = 0.0             * BRAW**2  + (   1.000000     )* BRAW
+ ( 0.0           )
   MINIMUM TWO SIGMA ERROR IS + OR -        1.07370
   MAXIMUM TWO SIGMA ERROR IS + OR -        1.34748




DEPRESS KEY 28 TO PAGE BACKWARD.
DEPRESS KEY 29 TO PAGE FORWARD.

DEPRESS KEY 30 TO VIEW OPTION TABLE
```

REPLY AREA

FIGURE 4.1.16

DAVIES EXAMPLE - FRAME 17

OUTPUT AREA

OPTION TABLE

DEPRESS KEY 1 TO PLOT THE PREDICTED RESPONSE AGAINST ONE FACTOR

DEPRESS KEY 2 TO PLOT A 95% CONFIDENCE INTERVAL AGAINST ONE FACTOR

DEPRESS KEY 3 TO PLOT CONTOURS

DEPRESS KEY 4 TO EDIT THE RAW DATA

DEPRESS KEY 5 TO RETURN TO THE RESPONSE EQUATION

REPLY AREA

FIGURE 4.1.17

DAVIES EXAMPLE - FRAME 18

```
THE FACTORS ARE AS FOLLOWS:
1. X1        **PLOTTING FACTOR**
2. X2


ENTER SEPARATED BY A COMMA THE MINIMUM AND MAXIMUM
VALUE TO BE PLOTTED.
```

FIGURE 4.1.18

DAVIES EXAMPLE - FRAME 19

```
THE FACTORS ARE AS FOLLOWS:

1. X1        -1.1
2. X2        -1.0.1


ENTER SEPARATED BY A COMMA THE MINIMUM AND MAXIMUM
VALUE TO BE PLOTTED.
UP TO FOUR PLOTS CAN BE VIEWED SIMULTANEOUSLY.
TO ACCOMPLISH THIS. SIMPLY TYPE MORE THAN ONE VALUE.
(SEPARATED BY COMMAS) WHEN ASKED TO ENTER THE FOLLOWING.
FIXED VALUES.
EXAMPLE:
10.5.15.?

PLEASE ENTER THE VALUE(S) FOR X2

DEPRESS KEY 1 TO PLOT.
```

FIGURE 4.1.19

DAVIES EXAMPLE - FRAME 20

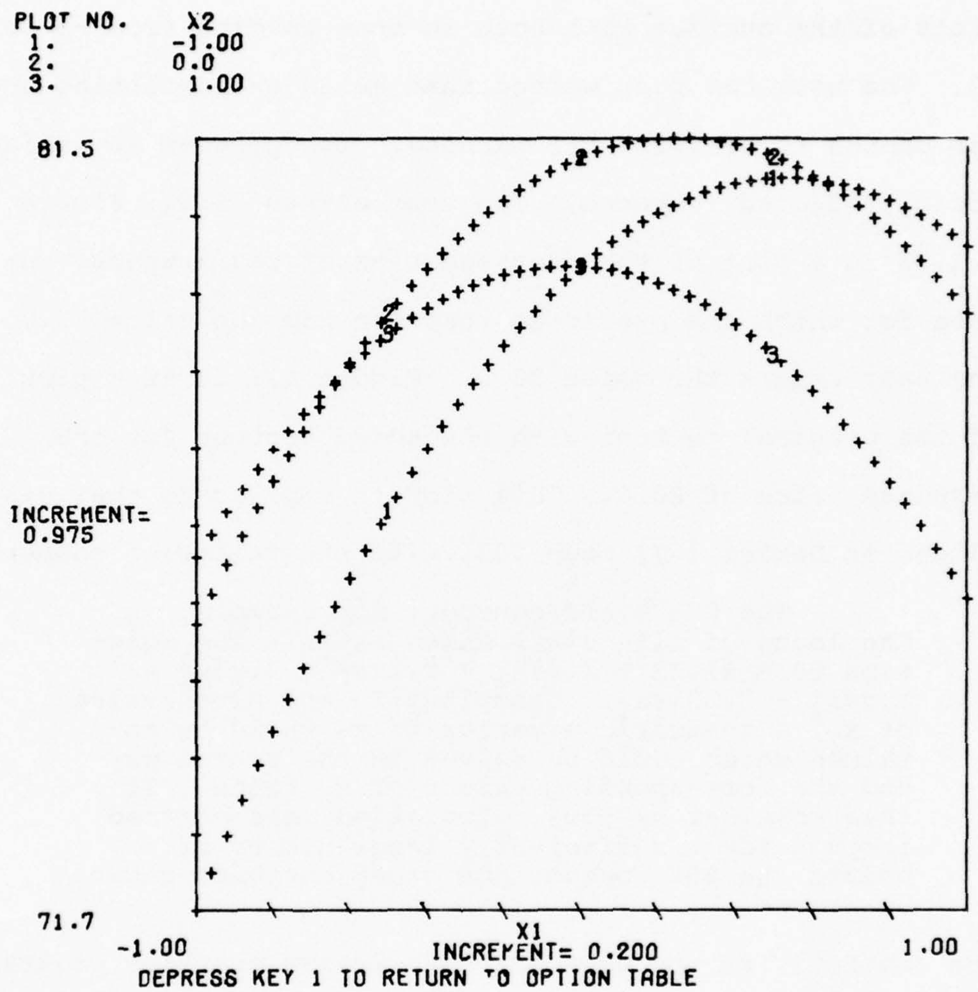


FIGURE 4.1.20

Figure 4.1.21 shows an analagous plot with X2 varying from -1 to +1 for values of X1 set at -1, 0 and +1.

In Figure 4.1.22, the user has requested contour plots of the surface with both factors ranging from -3 to +3. The user has been warned that he is extrapolating and has chosen to override the warning. The program is asking for a predicted response. The user enters 75.0. Figure 4.1.23 is a plot of the cross-section of the response surface for which the predicted response has the value 75.0. The user enters the value 80.0. Figure 4.1.24 is a plot of the original contour with the added contour for the response value of 80.0. This plot is similar to that displayed in Davies [7] page 523, with the following comment:

> The 80% yield contour, for example, is the locus of all points which satisfy the equation $80 = 81.22 + 1.97x_1 + 0.22x_2 - 3.93x_1^2 - 1.38x_2^2 - 2.22x_1x_2$. Substituting any given value of x_1, a quadratic equation in x_2 would be obtained which could be solved in the ususal way and the corresponding values of x_2 found. If this somewhat tedious calculation were carried through for a sufficiently large number of points the 80% contour and other contours could be drawn ...

The statistician who uses the interactive graphics program could perform the task in a few minutes, and the computer is put to work performing the "tedious task."

The user, seeking the maximum response, enters a value 82.0. The display in Figure 4.1.25 informs him that 82.0 is above the maximum of the surface (see the comment: NO SOLUTION FOR 82). The user tries the value 81.0.

DAVIES EXAMPLE - FRAME 21

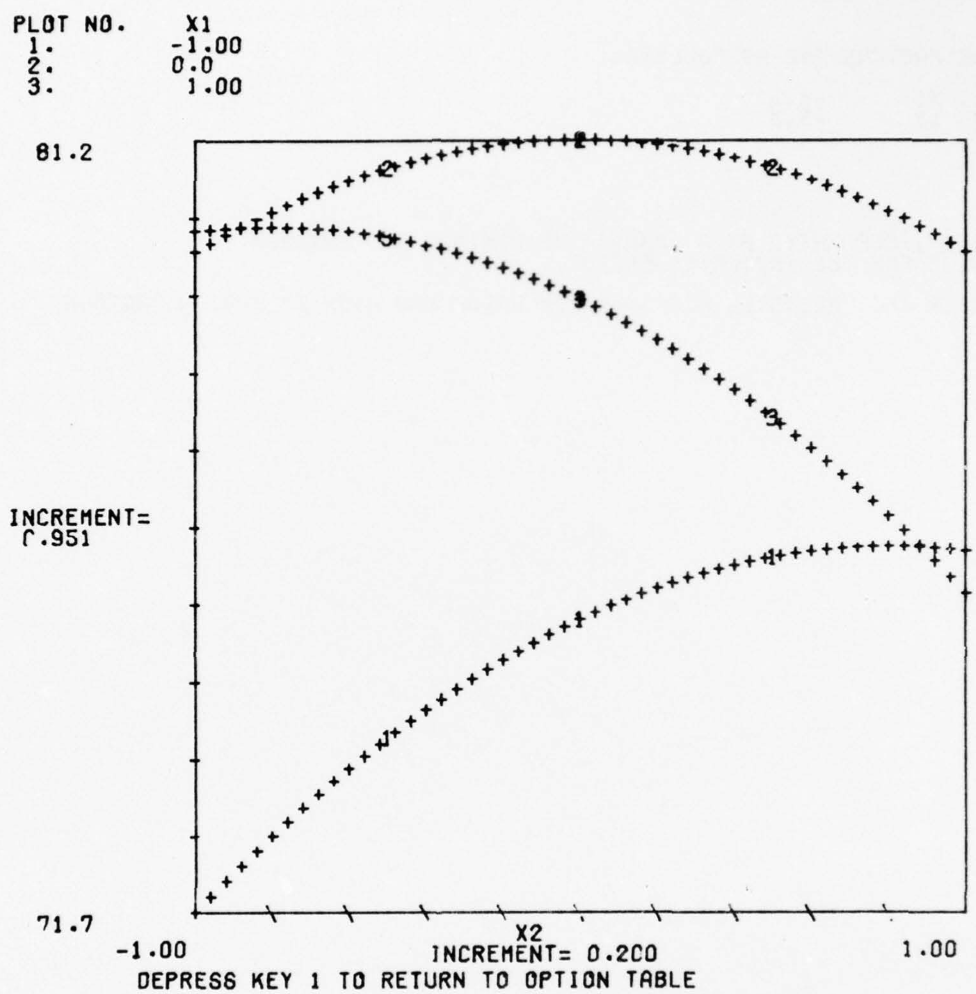


FIGURE 4.1.21

DAVIES EXAMPLE - FRAME 22

THE FACTORS ARE AS FOLLOWS:

```
 1. X1       -3.3
 2. X2       -3.3
```

ENTER, SEPARATED BY A COMMA, THE MINIMUM AND MAXIMUM
VALUE FOR THE INDICATED FACTOR

ENTER THE PREDICTED RESPONSE FOR WHICH YOU WISH TO PLOT A CONTOUR

FIGURE 4.1.22

DAVIES EXAMPLE - FRAME 23

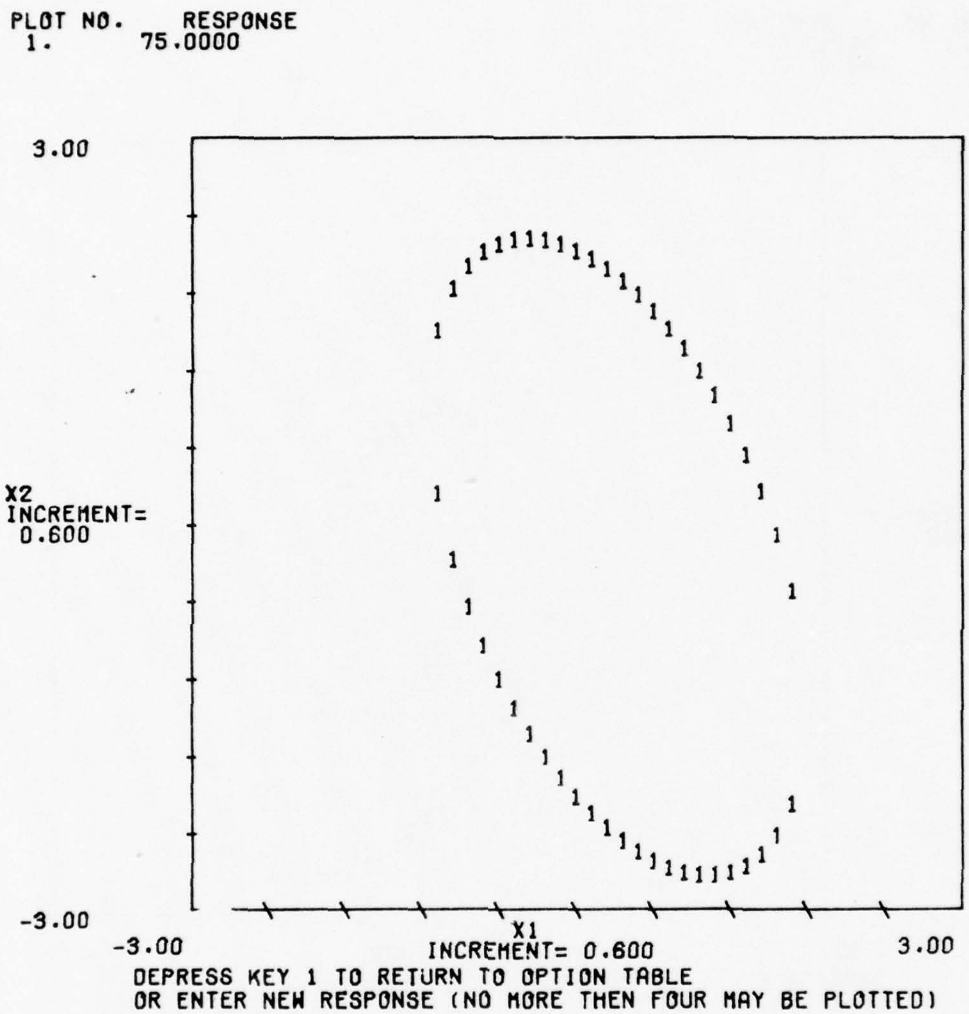


FIGURE 4.1.23

DAVIES EXAMPLE - FRAME 24

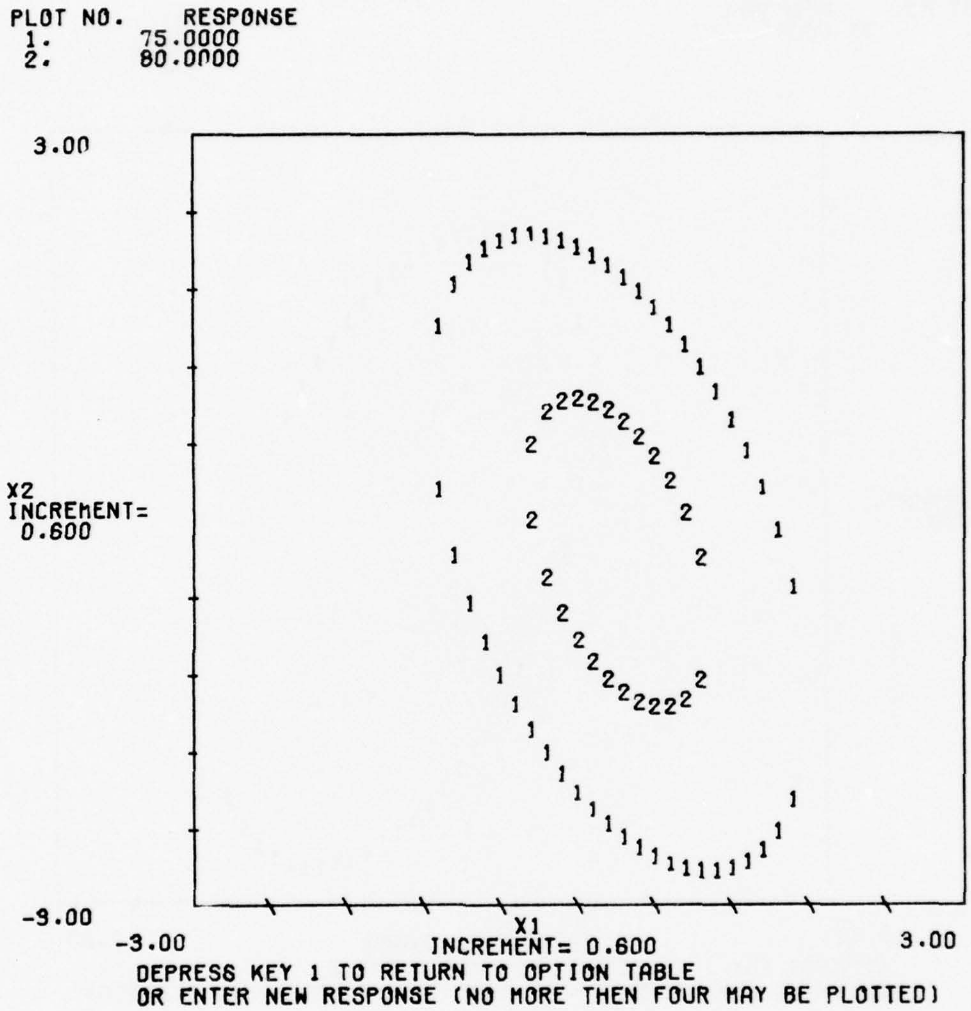


FIGURE 4.1.24

DAVIES EXAMPLE - FRAME 25

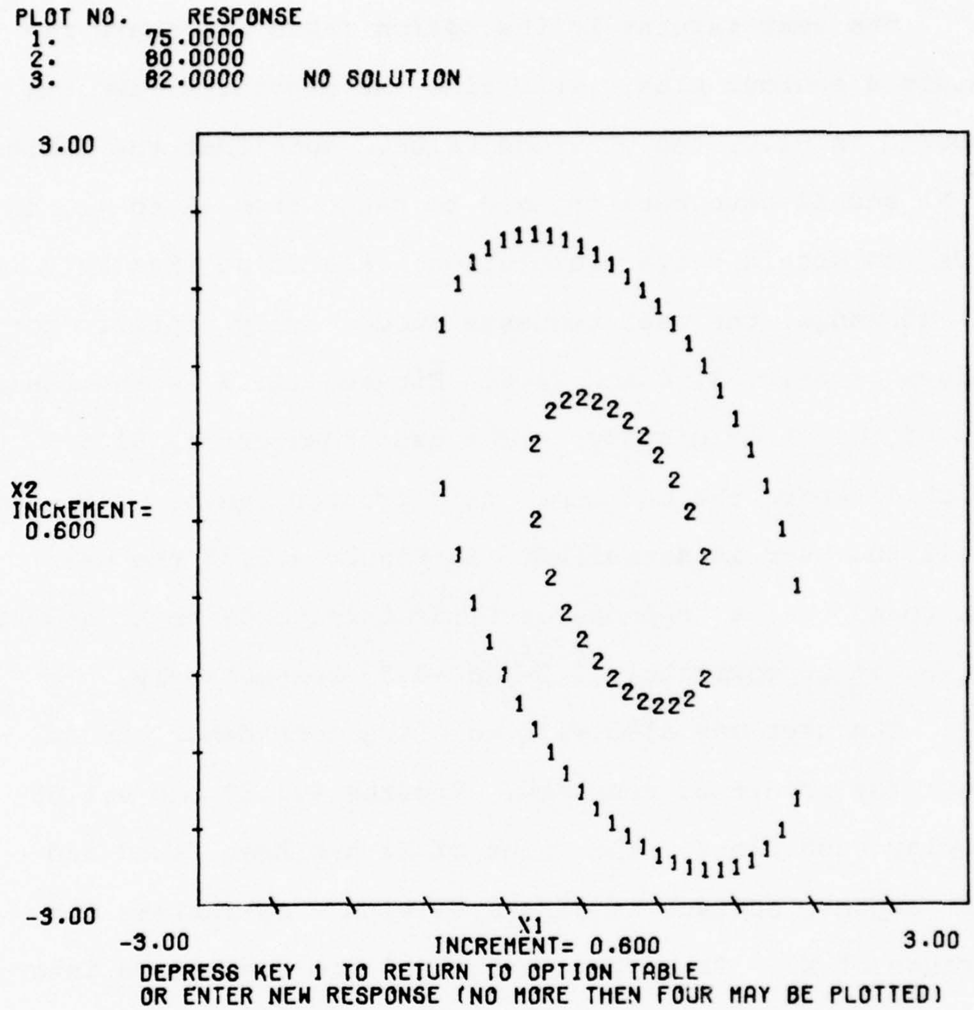


FIGURE 4.1.25

Figure 4.1.26 displays the inner contours for a response of 81.0. If further refinement is desired, a new series of contours will have to be initiated.

The user returns to the option table and again requests a contour plot. This time the predicted response entered is 81.0, the previous value. Note that the limits of X1 and X2 have been changed to range from -1 to +1, in order to obtain better resolution. Realizing that 82.0 is out of range, the user requests successively contours for values of 81.2, 81.4 and 81.6. Figure 4.1.27 is the final one of the three displays. The user then enters 81.6 which is above the maximum. This process can be continued until the user is satisfied. In Figure 4.1.28 the user has found that a response of 81.5 corresponds to X1 and X2 values of approximately 0.3 and -0.25 respectively.

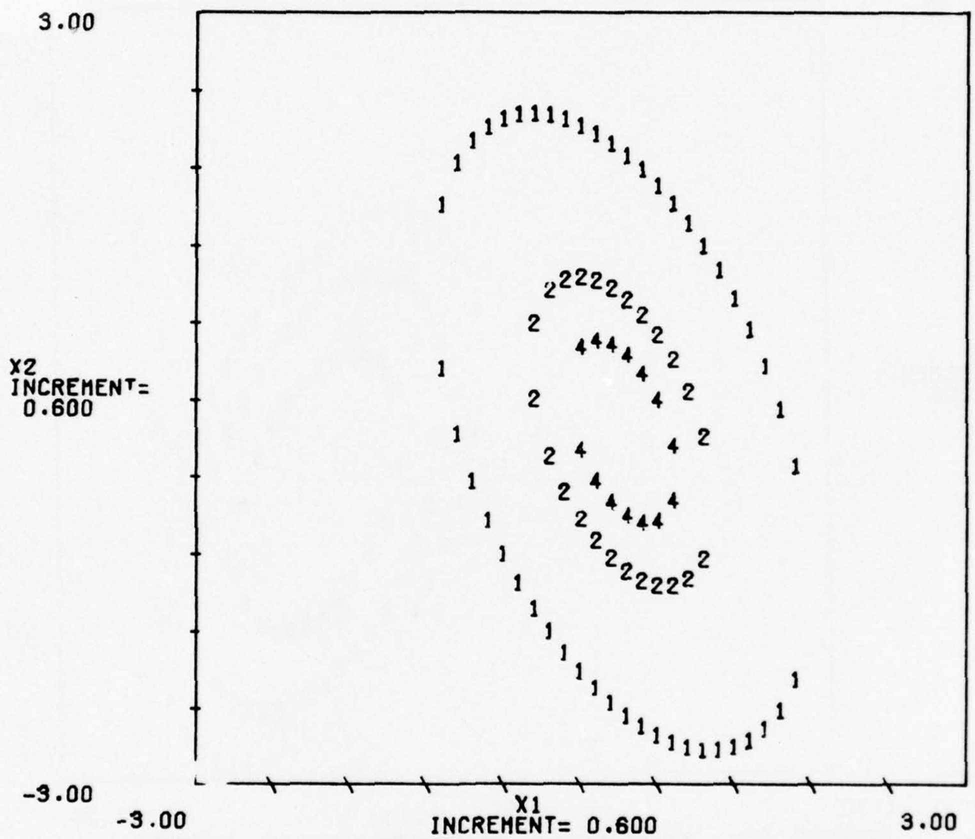
The user may also wish to place confidence bounds about the predicted response. Figures 4.1.29 and 4.1.30 display such plots. The value of X1 has been fixed and the response surface is displayed with + characters for a range of X2. The upper portion of the confidence interval is plotted with U characters and the lower, with L characters.

4.2 Process Evaluation

In many instances the investigator who conducts a factorial experiment is not interested in maximizing the

DAVIES EXAMPLE - FRAME 26

PLOT NO.	RESPONSE	
1.	75.0000	
2.	80.0000	
3.	82.0000	NO SOLUTION
4.	81.0000	

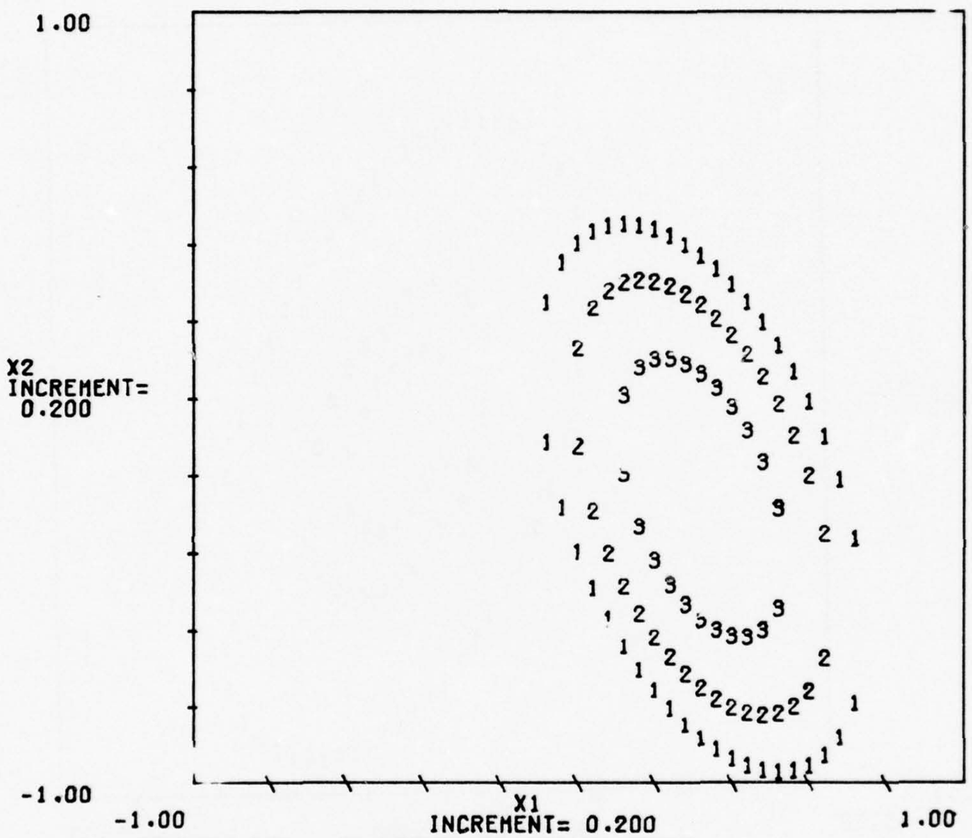


DEPRESS KEY 1 TO RETURN TO OPTION TABLE
OR ENTER NEW RESPONSE (NO MORE THEN FOUR MAY BE PLOTTED)

FIGURE 4.1.26

DAVIES EXAMPLE - FRAME 27

PLOT NO.	RESPONSE	
1.	81.0000	
2.	81.2000	
3.	81.4000	
4.	81.6000	NO SO'UTION



DEPRESS KEY 1 TO RETURN TO OPTION TABLE
OR ENTER NEW RESPONSE (NO MORE THEN FOUR MAY BE PLOTTED)

FIGURE 4.1.27

DAVIES EXAMPLE - FRAME 28

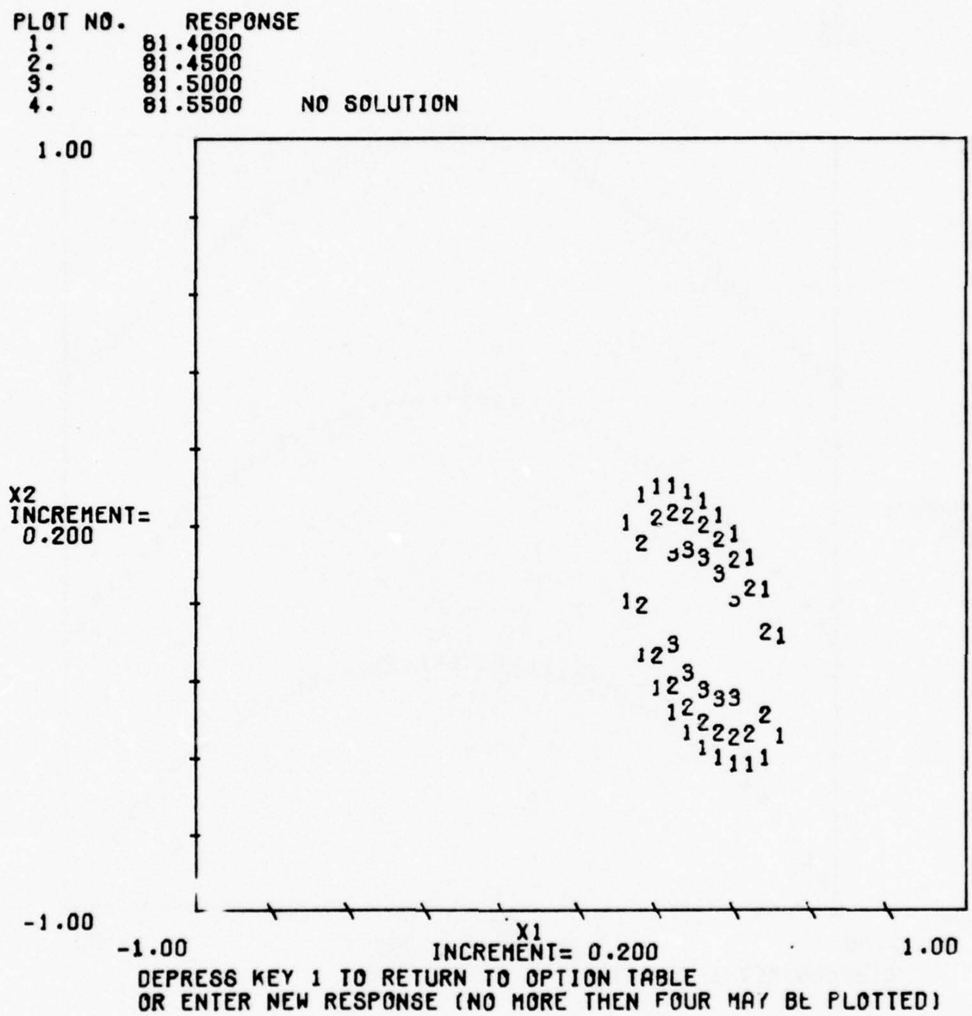


FIGURE 4.1.28

DAVIES EXAMPLE - FRAME 29

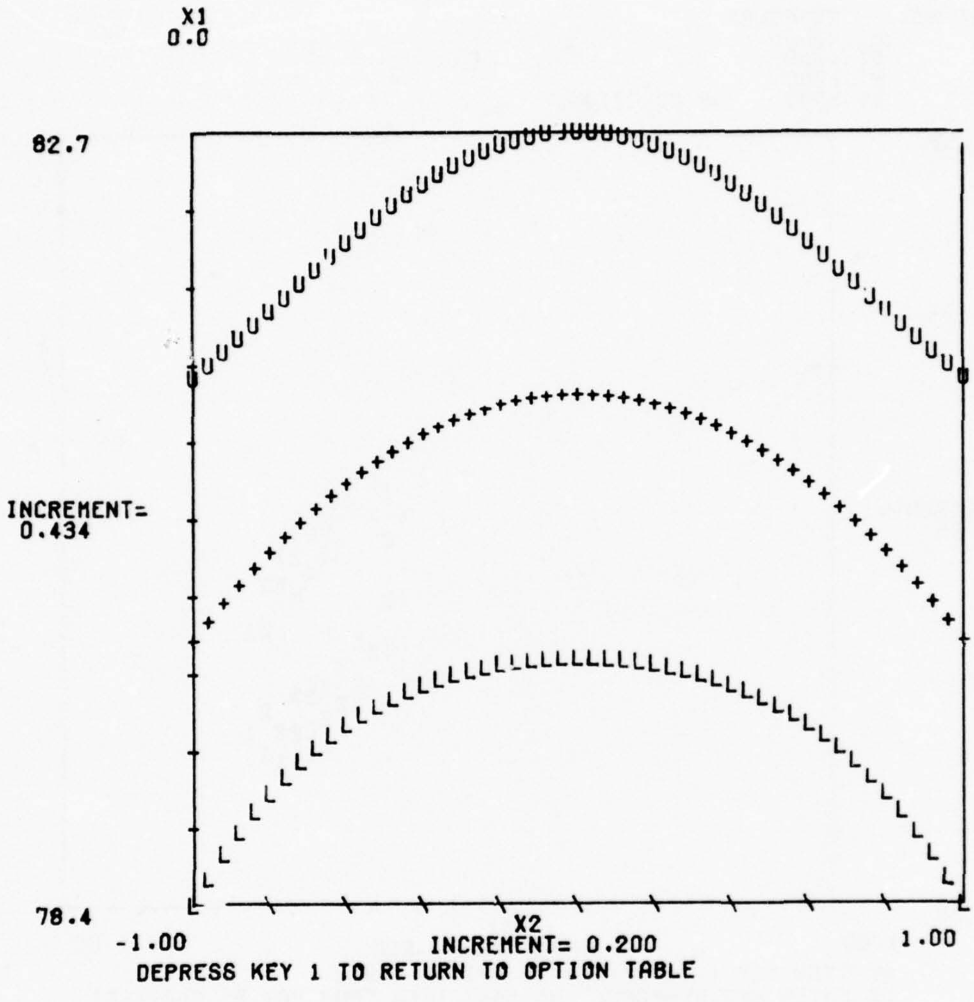


FIGURE 4.1.29

DAVIES EXAMPLE - FRAME 30

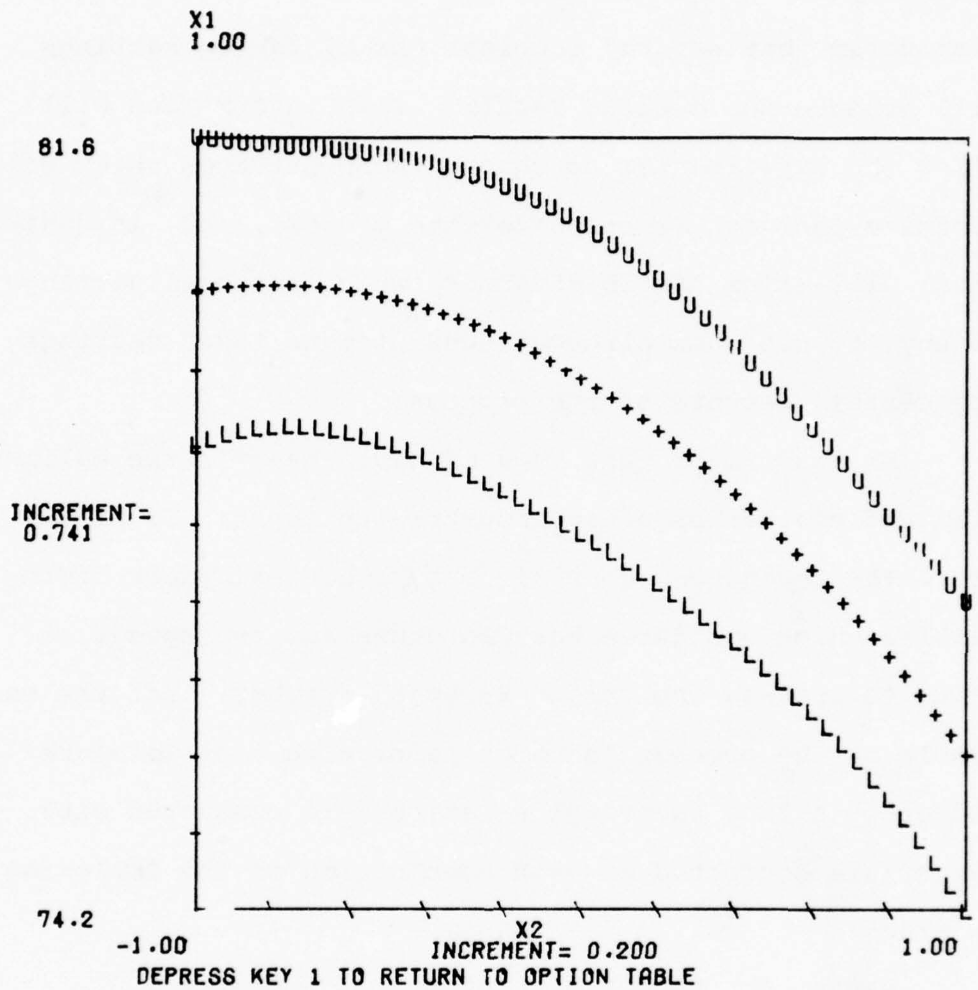


FIGURE 4.1.30

yield, but in uncovering those factor settings which produce certain intermediate levels of output. Further, if a certain yield is desired from the process, it might be discovered that several combinations of factor settings will produce the desired result. This information will allow the experimenter to choose those settings which will minimize the cost function for the process, and, in addition, will allow him to determine which residual settings, if any, to use when circumstances dictate fixed settings for certain factors of the process.

An example of this type of experiment is the calibration and evaluation of an industrial process. To illustrate the technique, consider a hypothetical paper drying machine which has three heating drums and two speeds at which to process the pulp. Suppose, further, that the end result of the process is to be paper with a 4% moisture content. A 2×3^3 factorial experiment is conducted with two trials conducted at each combination of the following settings:

SPEED	DRUM 1	DRUM 2	DRUM 3
1	70	100	120
2	100	120	130
	140	140	140

Drum temperatures are in degrees F.

Once the 108 observations of moisture content have been made, the researcher will wish to calculate and explore the response surface as in Figures 4.2.1 through

4.2.20.

In Figure 4.2.1, the user has indicated to the program that his experimental data consist of four factors with two replications. Note that, in this case, the program does not require an external estimate of the MSE. In Figure 4.2.2, the user has indicated that the first factor has two levels and that the remaining three have three levels each. In Figure 4.2.3, the program has been given the settings for the factors, e.g., the SPEED settings are 1 and 2, DRUM1 has settings of 70, 100 and 140 degrees, etc. In Figure 4.2.4, the user has begun to enter the observations. Line 1, which indicates all factors at level 1, refers to a SPEED of 1, DRUM1 at 70 degrees, DRUM2 at 100 degrees and DRUM3 at 120 degrees. In Figure 4.2.5, the user has completed the entry of the data for the 54 treatment combinations. The user then requests the program to perform the analysis. Figure 4.2.6 presents the mean squares to the user while Figure 4.2.7 presents the corresponding F values. Figure 4.2.8 presents the equation of the response surface with all of the factors present. In Figure 4.2.9 the user has requested a listing of the single degree of freedom sums of squares. He then requests the program to delete (set to 0) all coefficients of the orthogonal polynomials whose corresponding F ratios are less than 1.0. These sums of squares are added into the error. Figure 4.2.10 displays the simplified equation.

PROCESS EVALUATION EXAMPLE - FRAME 1

OUTPUT AREA

```
PLEASE ENTER THE NUMBER OF FACTORS.
4
PLEASE ENTER THE NUMBER OF REPLICATIONS.
2
DEPRESS KEY 1 TO CONTINUE.
```

REPLY AREA

FIGURE 4.2.1

PROCESS EVALUATION EXAMPLE - FRAME 2

```
                      OUTPUT AREA
THERE WILL BE AN OPPORTUNITY AT THE END OF THIS FRAME
TO CORRECT ANY INPUT ERRORS.
PLEASE ENTER THE NUMBER OF LEVELS FOR SPEED
FACTOR 1 HAS 2 LEVELS.

PLEASE ENTER THE NUMBER OF LEVELS FOR DRUM1
FACTOR 2 HAS 3 LEVELS.

PLEASE ENTER THE NUMBER OF LEVELS FOR DRUM2
FACTOR 3 HAS 3 LEVELS.

PLEASE ENTER THE NUMBER OF LEVELS FOR DRUM3
FACTOR 4 HAS 3 LEVELS.

DEPRESS KEY 1 TO CONTINUE OR KEY 2 TO RE-ENTER LEVEL
```

```
                      REPLY AREA
```

FIGURE 4.2.2

PROCESS EVALUATION EXAMPLE - FRAME 3

```
                              OUTPUT AREA
     IF YOU WISH THE LEVELS OF THE FACTORS TO BE
1 FOR LEVEL 1, 2 FOR LEVEL 2, ETC.  DEPRESS KEY 1

     IF YOU WISH TO ENTER VALUES FOR THE LEVELS, DEPRESS KEY 2

ENTER, SEPARATED BY COMMAS, THE LEVELS FOR SPEED

1,2

ENTER, SEPARATED BY COMMAS, THE LEVELS FOR DRUM1

70,100,140

ENTER, SEPARATED BY COMMAS, THE LEVELS FOR DRUM2

100,120,140

ENTER, SEPARATED BY COMMAS, THE LEVELS FOR DRUM3

120,130,140

DEPRESS KEY 1 TO CONTINUE
```

REPLY AREA

FIGURE 4.2.3

PROCESS EVALUATION EXAMPLE - FRAME 4

OUTPUT AREA

ENTER, SEPARATED BY COMMAS, ALL REPLICATIONS FOR THE FACTOR LEVELS INDICATED

A = SPEED B = DRUM1 C = DRUM2 D = DRUM3

DEPRESS KEY 30 TO EDIT A LINE OF DATA.

CELL	A	B	C	D	
1	1	1	1	1	4.8,5.2
2	2	1	1	1	9.8,10.2
3	1	2	1	1	4.4,4.6
4	2	2	1	1	9.4,9.6
5	1	3	1	1	4.3,4.4
6	2	3	1	1	9.3,9.4
7	1	1	2	1	4.5,4.7
8	2	1	2	1	9.5,9.7
9	1	2	2	1	4.2,4.5
10	2	2	2	1	9.2,9.5
11	1	3	2	1	3.8,4.0
12	2	3	2	1	8.8,9.0
13	1	1	3	1	4.6,4.7
14	2	1	3	1	9.6,9.7
15	1	2	3	1	

REPLY AREA

FIGURE 4.2.4

PROCESS EVALUATION EXAMPLE - FRAME 5

OUTPUT AREA

```
13    1 1 3 1 4.6,4.7
14    2 1 3 1 9.6,9.7
15    1 2 3 1 4.0,4.0
16    2 2 3 1 9.0,9.0
17    1 3 3 1 3.8,3.9
18    2 3 3 1 8.8,8.9
19    1 1 1 2 4.1,4.4
20    2 1 1 2 9.1,9.4
21    1 2 1 2 3.8,4.0
22    2 2 1 2 8.8,9.0
23    1 3 1 2 3.6,3.7
24    2 3 1 2 8.6,8.7
25    1 1 2 2 4.1,4.2
26    2 1 2 2 9.1,9.2
27    1 2 2 2 3.7,3.9
28    2 2 2 2 8.7,8.9
29    1 3 2 2 3.8,3.8
30    2 3 2 2 8.8,8.8
31    1 1 3 2 4.0,4.4
32    2 1 3 2 9.0,9.4
33    1 2 3 2 3.6,3.8
34    2 2 3 2 8.6,8.8
35    1 3 3 2 3.5,3.6
36    2 3 3 2 8.5,8.6
37    1 1 1 3 4.2,4.3
38    2 1 1 3 9.2,9.3
39    1 2 1 3 3.5,3.7
40    2 2 1 3 8.5,8.7
41    1 3 1 3 3.4,3.6
42    2 3 1 3 8.4,8.6
43    1 1 2 3 3.9,4.0
44    2 1 2 3 8.9,9.0
45    1 2 2 3 3.1,3.3
46    2 2 2 3 8.1,8.3
47    1 3 2 3 3.0,3.1
48    2 3 2 3 8.0,8.1
49    1 1 3 3 3.4,3.5
50    2 1 3 3 8.4,8.5
51    1 2 3 3 2.9,3.1
52    2 2 3 3 7.9,8.1
53    1 3 3 3 2.8,2.9
54    2 3 3 3 7.8,7.9
DATA ENTRY IS COMPLETE.
DEPRESS KEY 1 TO PERFORM ANALYSIS
DEPRESS KEY 2 TO REVIEW OR EDIT DATA.
```

REPLY AREA

FIGURE 4.2.5

PROCESS EVALUATION EXAMPLE - FRAME 6

OUTPUT AREA

ANALYSIS OF VARIANCE

SOURCE OF VARIATION					DF	MEAN SQUARES
SPEED	*				1	674.997
DRUM1	*				2	4.31147
SPEED	*DRUM1	*			2	0.763596E-11
DRUM2	*				2	1.57813
SPEED	*DRUM2	*			2	0.258701E-10
DRUM1	*DRUM2	*			4	0.537025E-02
SPEED	*DRUM1	*DRUM2	*		4	0.485695E-11
DRUM3	*				2	7.74704
SPEED	*DRUM3	*			2	0.646752E-11
DRUM1	*DRUM3	*			4	0.609255E-01
SPEED	*DRUM1	*DRUM3	*		4	0.231795E-11
DRUM2	*DRUM3	*			4	0.265826
SPEED	*DRUM2	*DRUM3	*		4	0.323376E-11
DRUM1	*DRUM2	*DRUM3	*		8	0.548151E-01
SPEED	*DRUM1	*DRUM2	*DRUM3	*	8	0.423799E-11
ALL EFFECTS					53	13.2837
ERROR					54	0.187806E-01

DEPRESS KEY 1 FOR SUMS OF SQUARES.
DEPRESS KEY 2 FOR MEAN SQUARES.
DEPRESS KEY 3 FOR F VALUES.

DEPRESS KEY 30 TO VIEW OPTION TABLE

REPLY AREA

FIGURE 4.2.6

PROCESS EVALUATION EXAMPLE - FRAME 7

OUTPUT AREA

ANALYSIS OF VARIANCE

SOURCE OF VARIATION					DF	F
SPEED	*				1	0.3594E 05
DRUM1	*				2	229.6
SPEED	*DRUM1	*			2	0.4066E-09
DRUM2	*				2	84.03
SPEED	*DRUM2	*			2	0.1377E-08
DRUM1	*DRUM2	*			4	0.2859
SPEED	*DRUM1	*DRUM2	*		4	0.2588E-09
DRUM3	*				2	412.5
SPEED	*DRUM3	*			2	0.3444E-09
DRUM1	*DRUM3	*			4	3.244
SPEED	*DRUM1	*DRUM3	*		4	0.1234E-09
DRUM2	*DRUM3	*			4	14.16
SPEED	*DRUM2	*DRUM3	*		4	0.1722E-09
DRUM1	*DRUM2	*DRUM3	*		8	2.919
SPEED	*DRUM1	*DRUM2	*DRUM3	*	8	0.2257E-09
ALL EFFECTS					53	707.3
ERROR					54	

DEPRESS KEY 1 FOR SUMS OF SQUARES.
DEPRESS KEY 2 FOR MEAN SQUARES.
DEPRESS KEY 3 FOR F VALUES.

DEPRESS KEY 30 TO VIEW OPTION TABLE

REPLY AREA

FIGURE 4.2.7

PROCESS EVALUATION EXAMPLE - FRAME 8

```
                        OUTPUT AREA
               FACTOR*A    IS  SPEED
               FACTOR*B    IS  DRUM1
               FACTOR*C    IS  DRUM2
               FACTOR*D    IS  DRUM3

      IN STANDARD FACTOR VALUES, Y =
+  6.299990        +  2.499993     *A
-0.1749993    *B  + 0.1749992     *B**2
-0.9999847E-01*C  -0.1250076E-01*B*C
+ 0.6249999E-01*B**2 *C  -0.1375007     *B*C**2
-0.6249996E-01*B**2 *C**2   -0.5750006     *D
-0.4999923E-01*B*D  + 0.2000006     *B**2 *D
-0.2500248E-01*C*D  + 0.3750038E-01*B*C*D
-0.4999638E-01*B**2 *C*D  + 0.1000013     *C**2 *D
+ 0.6249809E-01*B*C**2 *D  -0.2000016     *B**2 *C**2 *D
-0.2499621E-01*D**2   -0.2250021     *B*D**2
-0.7500076E-01*B**2 *D**2   -0.1750001     *C*D**2
+ 0.1250076E-01*B*C*D**2   -0.7499975E-01*B**2 *C*D**2
+ 0.1875038     *B*C**2 *D**2   + 0.1750001     *B**2 *C**2 *D**2

   WHERE*A    =  2.000000      * ARAW  + ( -3.000000      )
   WHERE*B    =-0.1190476E-03 * BRAW**2  + ( 0.5357143E-01)* BRAW
+ ( -4.166665      )
   WHERE*C    = 0.0              * CRAW**2  + ( 0.5000000E-01)* CRAW
+ ( -5.999999      )
   WHERE*D    = 0.0              * DRAW**2  + ( 0.9999996E-01)* DRAW
+ ( -13.00000      )
   MINIMUM TWO SIGMA ERROR IS + OR -          0.27535
   MAXIMUM TWO SIGMA ERROR IS + OR -          0.33620




DEPRESS KEY 28 TO PAGE BACKWARD
DEPRESS KEY 29 TO PAGE FORWARD

DEPRESS KEY 30 TO VIEW OPTION TABLE
```

REPLY AREA

FIGURE 4.2.8

PROCESS EVALUATION EXAMPLE - FRAME 9

OUTPUT AREA

SINGLE DEGREE OF FREEDOM DISPLAY

DEPRESS KEY 1 TO DELETE TERMS BY CELL NUMBER
DEPRESS KEY 2 TO DELETE TERMS BY F COMPARISON
DEPRESS KEY 28 TO PAGE BACKWARD
DEPRESS KEY 29 TO PAGE FORWARD

DEPRESS KEY 30 TO VIEW OPTION TABLE

MSE = 0.18780589E-01

A = SPEED B = DRUM1 C = DRUM2 D = DRUM3

CELL	A	B	C	D	COEFFICIENT	SUM OF SQUARES	F RATIO
31	0	0	2	1	-0.11111047E-01	0.17777566E-01	0.94659251
32	1	0	2	1	0.10596381E-06	0.16168794E-11	0.86093105E-10
33	0	1	2	1	0.20832695E-01	0.41664124E-01	2.2184668
34	1	1	2	1	-0.28808910E-06	0.79675520E-11	0.42424397E-09
35	0	2	2	1	-0.22222411E-01	0.14222461	7.5729570
36	1	2	2	1	-0.52981903E-07	0.80843969E-12	0.43046552E-10
37	0	0	0	2	0.92683011E-03	0.18554705E-03	0.98797232E-02
38	1	0	0	2	0.21192761E-06	0.97012762E-11	0.51655857E-09
39	0	1	0	2	-0.33333246E-01	0.15999907	8.5193844
40	1	1	0	2	0.59604645E-07	0.51159077E-12	0.27240390E-10
41	0	2	0	2	0.46295561E-02	0.92589743E-02	0.49300766
42	1	2	0	2	-0.10596381E-06	0.48506381E-11	0.25827918E-09
43	0	0	1	2	-0.74999988E-01	0.81000042	43.129654
44	1	0	1	2	0.10596381E-06	0.16168794E-11	0.86093105E-10
45	0	1	1	2	0.41669197E-02	0.16668700E-02	0.88754892E-01
46	1	1	1	2	0.14901161E-06	0.21316282E-11	0.11350167E-09
47	0	2	1	2	-0.83333105E-02	0.19999895E-01	1.0649233
48	1	2	1	2	-0.52981903E-07	0.80843969E-12	0.43046552E-10
49	0	0	2	2	0.12962870E-01	0.72591484E-01	3.8652391
50	1	0	2	2	-0.10596381E-06	0.48506381E-11	0.25827918E-09
51	0	1	2	2	0.20833757E-01	0.12500507	6.6560774
52	1	1	2	2	0.22848445E-06	0.15035084E-10	0.80056495E-09
53	0	2	2	2	0.64814873E-02	0.36296368E-01	1.9326525
54	1	2	2	2	-0.52981903E-07	0.24253186E-11	0.12913964E-09

REPLY AREA

FIGURE 4.2.9

PROCESS EVALUATION EXAMPLE - FRAME 10

```
                         OUTPUT AREA
                FACTOR*A   IS  SPEED
                FACTOR*B   IS  DRUM1
                FACTOR*C   IS  DRUM2
                FACTOR*D   IS  DRUM3

       IN STANDARD FACTOR VALUES, Y =
+  6.259251        +  2.499995      *A
-0.1833315     *B  + 0.2388877     *B**2
-0.9166509E-01*C  + 0.4999986E-01*B**2 *C
+ 0.3611118E-01*C**2   -0.1250025     *B*C**2
-0.1166667     *B**2 *C**2   -0.5972227     *D
-0.4166539E-01*B*D  + 0.2000006     *B**2 *D
-0.2500248E-01*C*D  + 0.3750038E-01*B*C*D
-0.4999638E-01*B**2 *C*D  + 0.1333345     *C**2 *D
+ 0.6249809E-01*B*C**2 *D  -0.2000016     *B**2 *C**2 *D
-0.2250021     *B*D**2    -0.1166667     *B**2 *D**2
-0.1750001     *C*D**2    -0.7499975E-01*B**2 *C*D**2
+ 0.1875038     *B*C**2 *D**2   + 0.1750001     *B**2 *C**2 *D**2

   WHERE*A   =  2.000000      * ARAW  + ( -3.000000     )
   WHERE*B   =-0.1190476E-03 * BRAW**2  + ( 0.5357143E-01)* BRAW
+ ( -4.166665     )
   WHERE*C   = 0.0           * CRAW**2  + ( 0.5000000E-01)* CRAW
+ ( -5.999999     )
   WHERE*D   = 0.0           * DRAW**2  + ( 0.9999996E-01)* DRAW
+ ( -13.00000     )
   MINIMUM TWO SIGMA ERROR IS + OR -        0.22009
   MAXIMUM TWO SIGMA ERROR IS + OR -        0.23861




DEPRESS KEY 28 TO PAGE BACKWARD.
DEPRESS KEY 29 TO PAGE FORWARD.

DEPRESS KEY 30 TO VIEW OPTION TABLE
```

REPLY AREA

FIGURE 4.2.10

At this point, the user is in a position to explore the response surface. Figure 4.2.11 displays SPEED against the predicted response for the factor settings indicated above the plot. Note that as DRUM3 is varied from 120 degrees to 140 degrees, the moisture of the resultant paper is reduced by an almost constant value of 0.6%. The principal result of the frame is that a speed of 2 produces a product which is much too wet. In Figure 4.2.12 it has been decided to limit the speed to 1 (slow) and to plot the predicted response against the settings of DRUM1 with DRUM2 varying through its range of setting while DRUM3 remains at its lowest setting. The first curve does not get down to 4%; however, a DRUM1 setting of about 133 degrees produces the desired result for curve 2, while only 105 degrees is required by curve 3. In summary, this plot indicates that 4% moisture content can be expected at the following two settings:

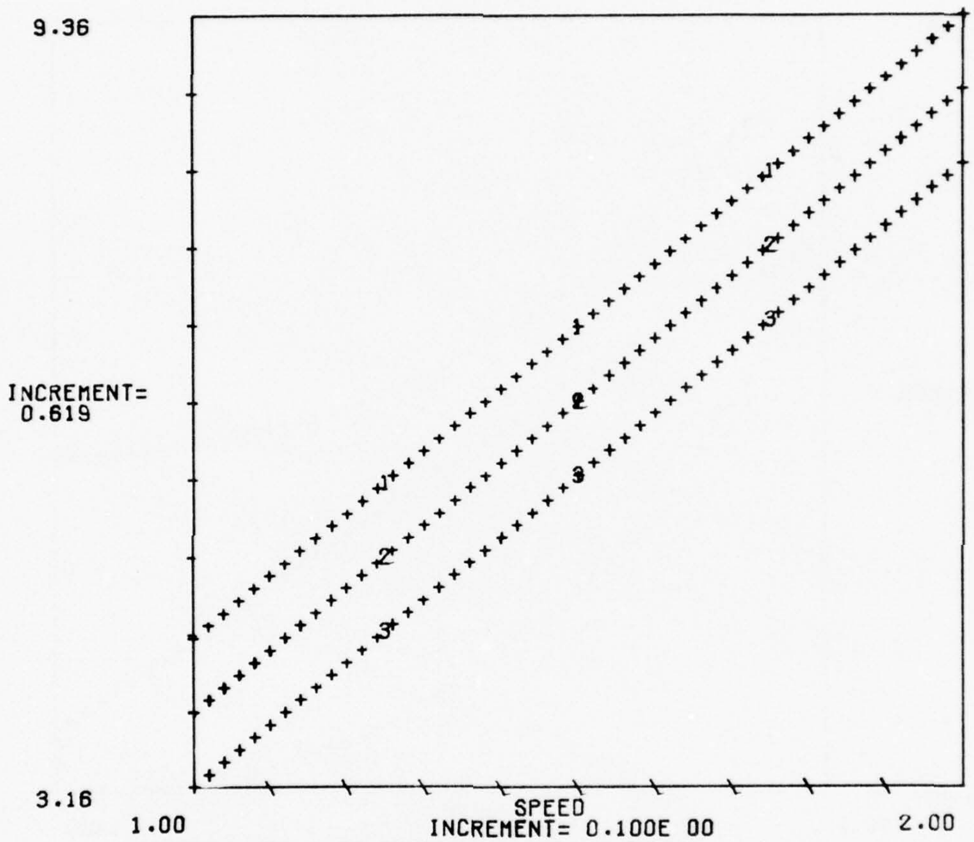
SPEED	DRUM 1	DRUM 2	DRUM 3
1	133	120	120
1	105	140	120

The user may thus investigate the surface in this fashion searching for reasonable levels.

Figure 4.2.13 illustrates a phenomenon that can occur when using polynomials for purposes of interpolation. Curve 2 appears to show that as the temperature of DRUM1 is increased, the resulting paper is wetter. Upon consid-

PROCESS EVALUATION EXAMPLE - FRAME 11

PLOT NO.	DRUM1	DRUM2	DRUM3
1.	100.	120.	120.
2.	100.	120.	130.
3.	100.	120.	140.



DEPRESS KEY 1 TO RETURN TO OPTION TABLE

FIGURE 4.2.11

PROCESS EVALUATION EXAMPLE - FRAME 12

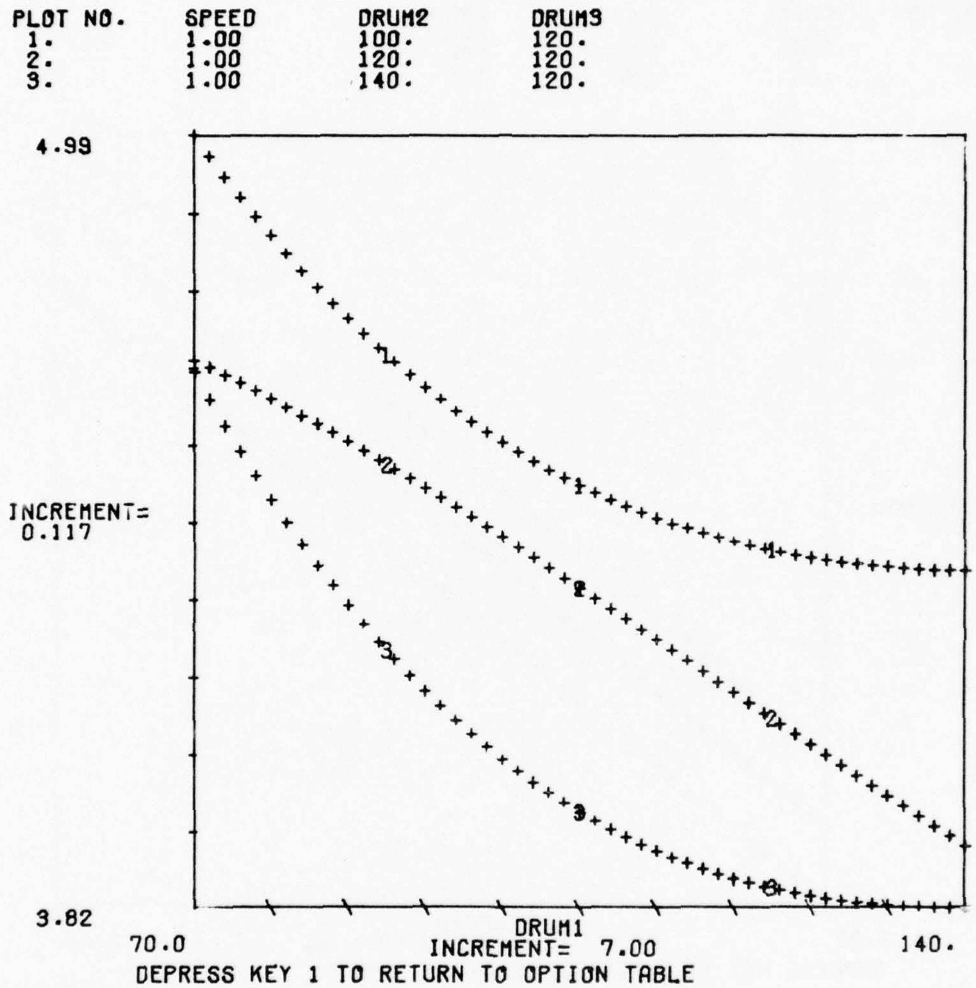
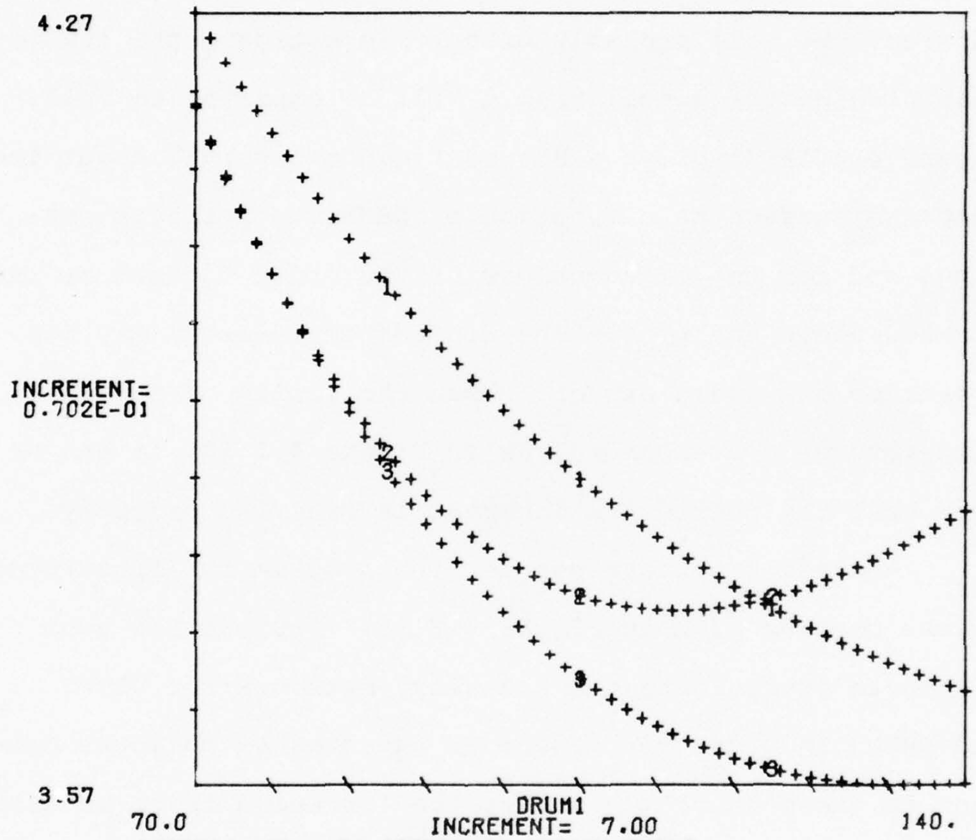


FIGURE 4.2.12

PROCESS EVALUATION EXAMPLE - FRAME 13

PLOT NO.	SPEED	DRUM2	DRUM3
1.	1.00	100.	130.
2.	1.00	120.	130.
3.	1.00	140.	130.



DEPRESS KEY 1 TO RETURN TO OPTION TABLE

FIGURE 4.2.13

eration, it is apparent that the predicted response drops to a constant level in the midrange of the DRUM1 settings and that the quadratic required to fit the data reaches a minimum and begins an upswing - a mathematical artifact.

After the user has decided upon certain settings of interest, he will probably wish to investigate the limits within which the actual result will be expected to fall. Figure 4.2.14 displays a 95% confidence interval about the response surface as a function of DRUM1 temperature settings and for the fixed values of the other factors as indicated above the plot. The confidence interval may not appear as one would expect. When the limits of DRUM1 temperatures are expanded, as in Figure 4.2.15, it can be seen that the confidence interval is behaving properly.

Another important use for the program is illustrated by the contour plot in Figure 4.2.16. Suppose the user wishes to investigate the necessary settings for DRUM2 and DRUM3 in order to produce an expected 4% moisture content in the product given that the low speed is to be used and that DRUM1 is to be set at 100 degrees. The curve displayed is the locus of all settings satisfying this condition. The user may now make his selection from this subset. Figures 4.2.17 through 4.2.19 display other contours which would be of interest to the user. Figure 4.2.20 represents contours of the surface for various predicted responses. Note the complexity of the projections even

PROCESS EVALUATION EXAMPLE - FRAME 14

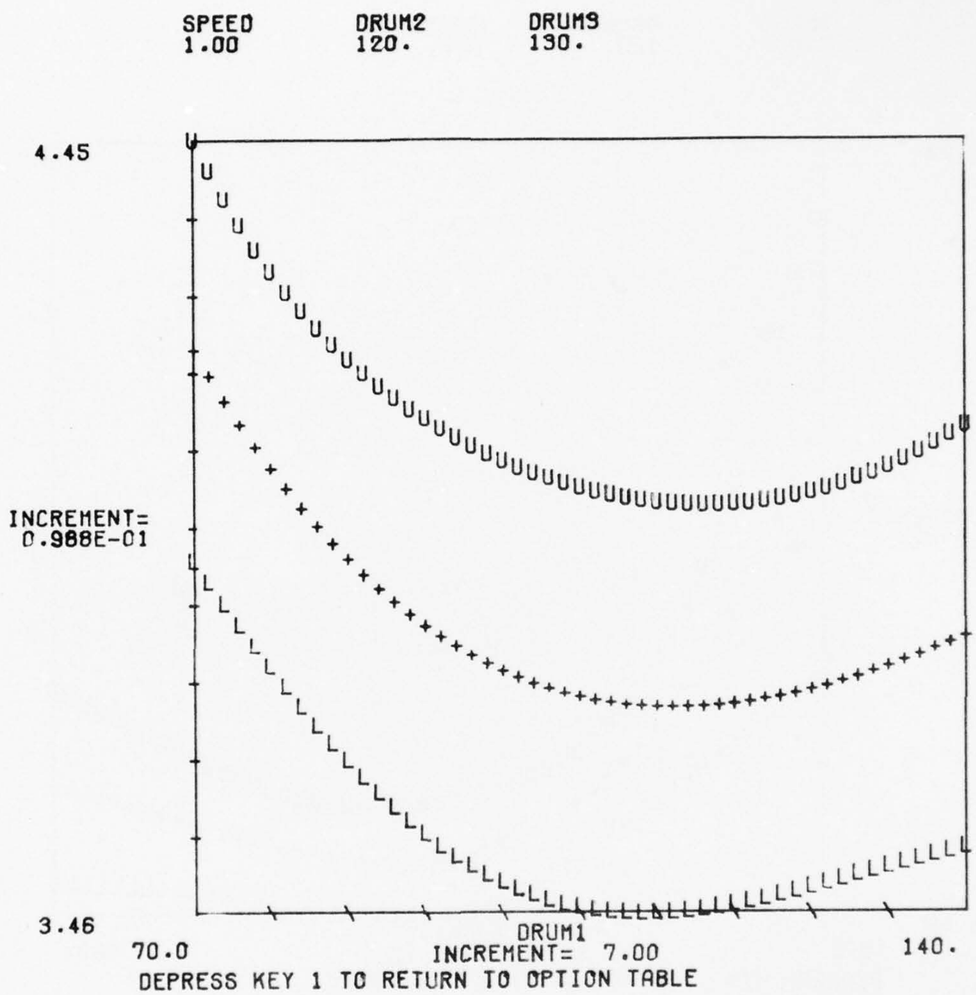


FIGURE 4.2.14

PROCESS EVALUATION EXAMPLE - FRAME 15

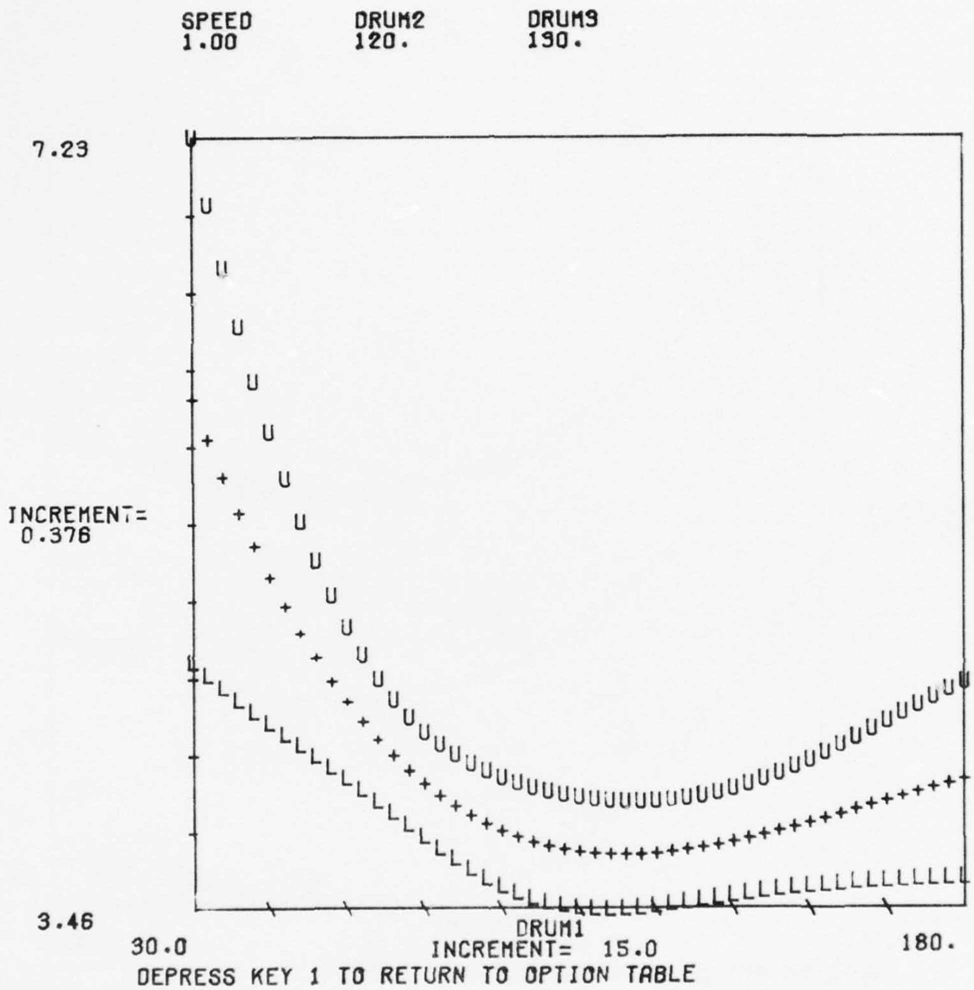
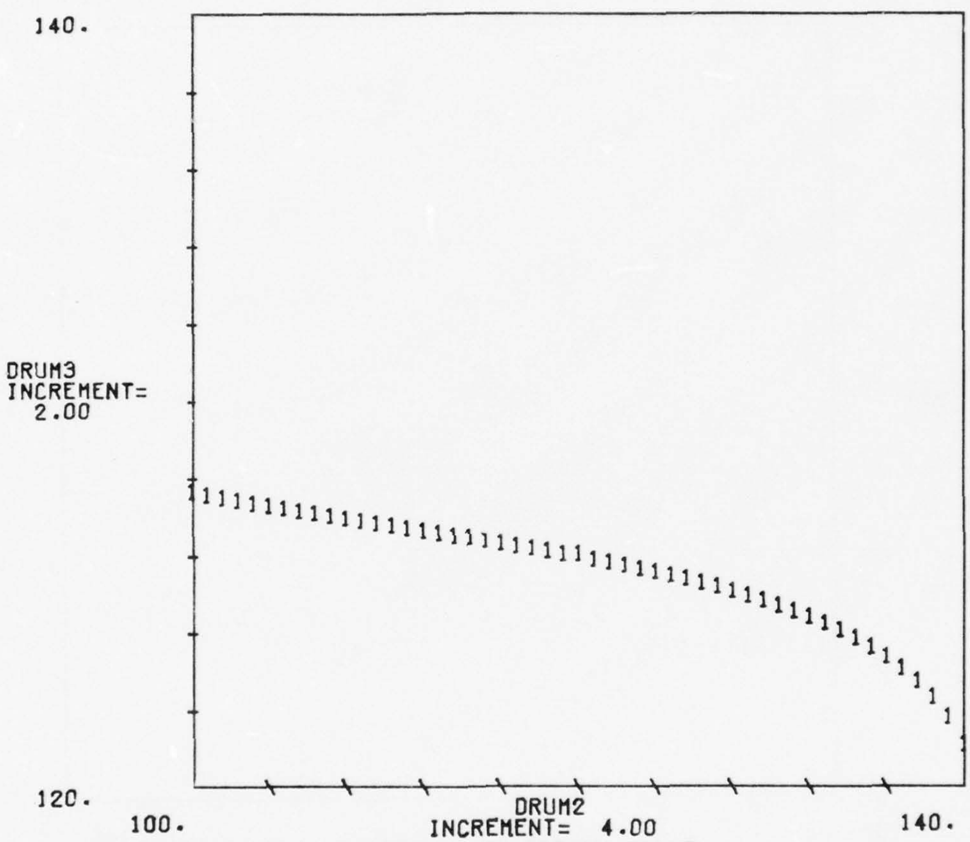


FIGURE 4.2.15

PROCESS EVALUATION EXAMPLE - FRAME 16

PLOT NO.	RESPONSE	SPEED	DRUM1
1.	4.00000	1.00000	100.000

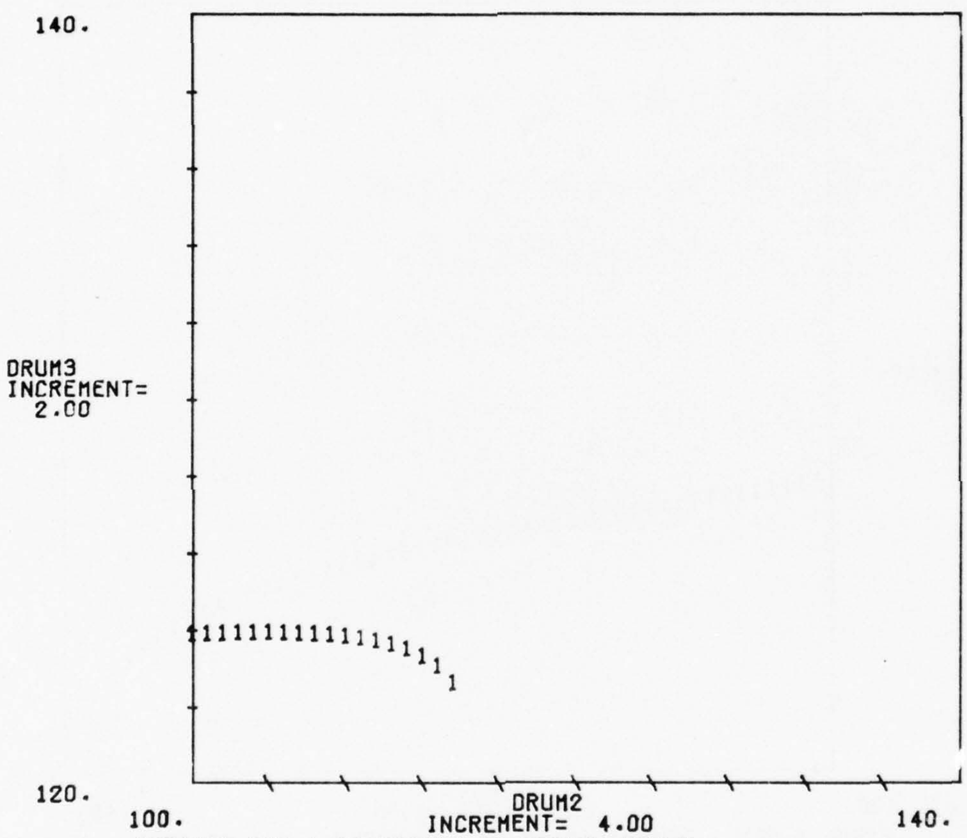


DEPRESS KEY 1 TO RETURN TO OPTION TABLE
OR ENTER NEW RESPONSE (NO MORE THEN FOUR MAY BE PLOTTED)

FIGURE 4.2.16

PROCESS EVALUATION EXAMPLE - FRAME 17

PLOT NO. RESPONSE SPEED DRUM1
1. 4.00000 1.00000 140.000



DEPRESS KEY 1 TO RETURN TO OPTION TABLE
OR ENTER NEW RESPONSE (NO MORE THEN FOUR MAY BE PLOTTED)

FIGURE 4.2.17

PROCESS EVALUATION EXAMPLE - FRAME 18

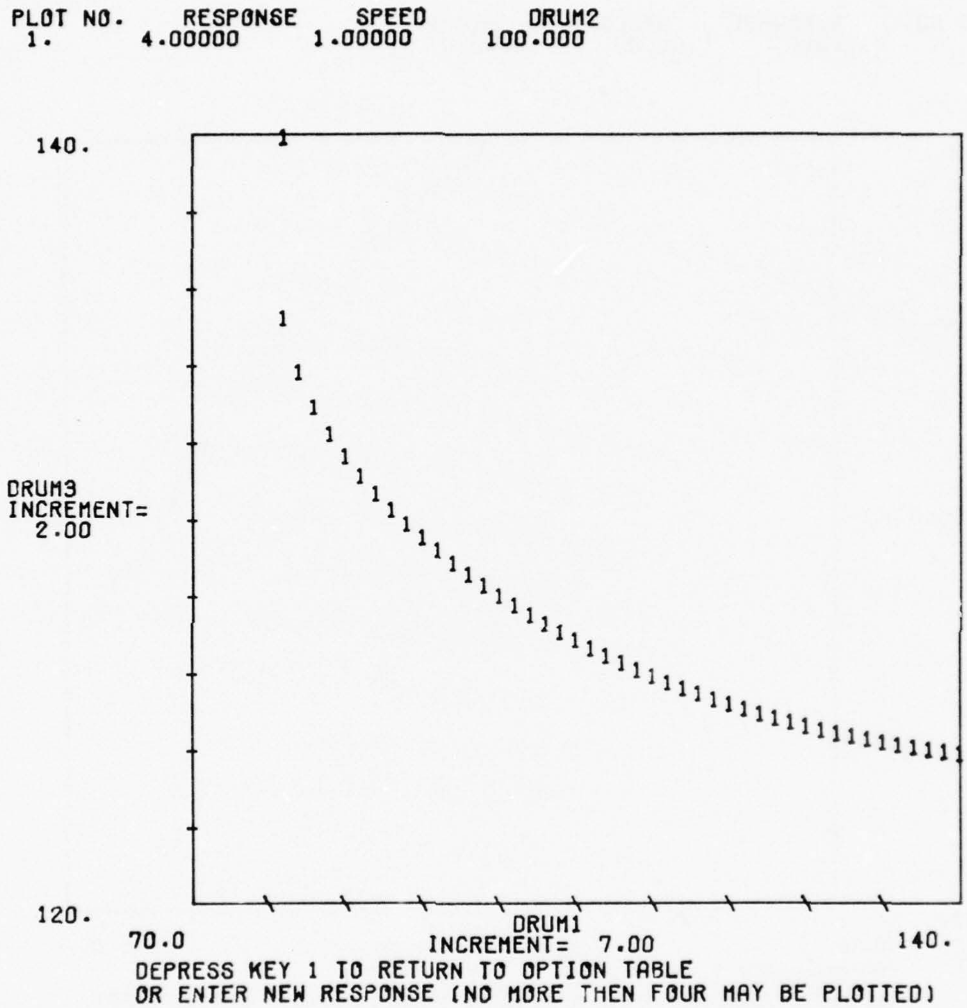
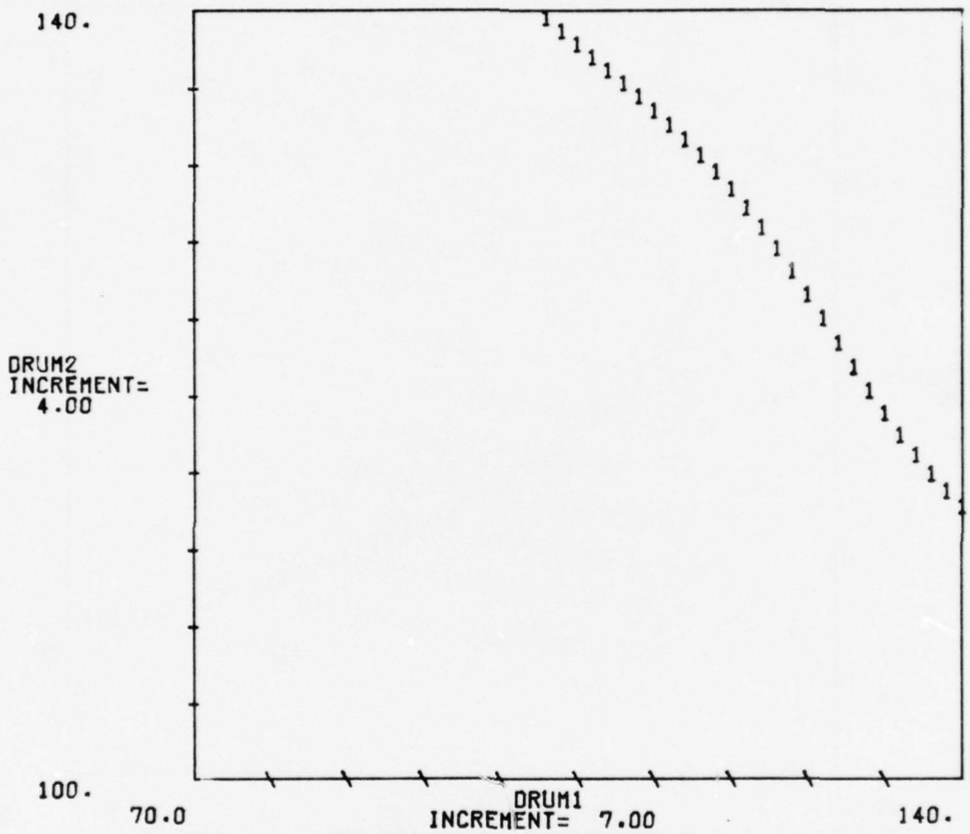


FIGURE 4.2.18

PROCESS EVALUATION EXAMPLE - FRAME 19

PLOT NO.	RESPONSE	SPEED	DRUMS
1.	4.00000	1.00000	120.000

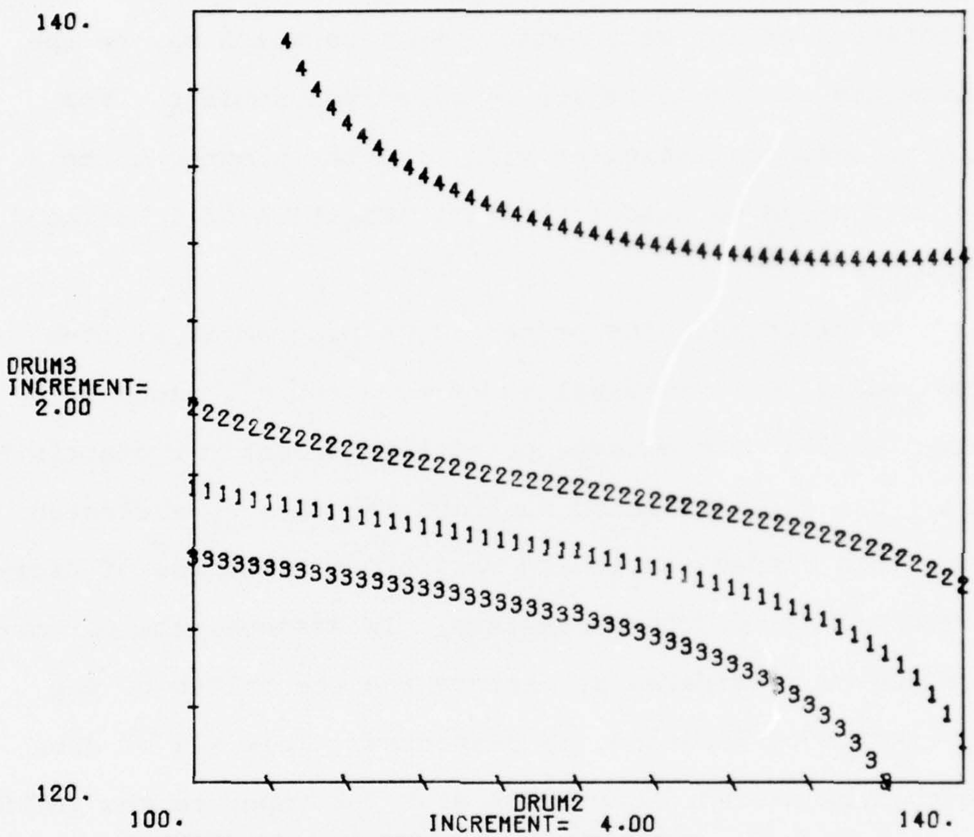


DEPRESS KEY 1 TO RETURN TO OPTION TABLE
OR ENTER NEW RESPONSE (NO MORE THEN FOUR MAY BE PLOTTED)

FIGURE 4.2.19

PROCESS EVALUATION EXAMPLE - FRAME 20

PLOT NO.	RESPONSE	SPEED	DRUM1
1.	4.00000	1.00000	100.000
2.	3.90000	1.00000	100.000
3.	4.10000	1.00000	100.000
4.	3.50000	1.00000	100.000



DEPRESS KEY 1 TO RETURN TO OPTION TABLE
OR ENTER NEW RESPONSE (NO MORE THEN FOUR MAY BE PLOTTED)

FIGURE 4.2.20

though the responses are all within 0.6 of each other.

4.3 Likelihood Function Contours

Although the YATES program was designed initially for the analysis of response surfaces, it is useful for the exploration of any mathematical surface which may be approximated over some region by mixed polynomials. For this reason, statisticians will find the program to be a tool which can be used in the investigation of likelihood functions.

To illustrate the procedure, a program is written which would, for given values of ν, α and σ^2, generate a random sample from a three parameter Student's t distribution. The log-likelihood function can then be evaluated using this random sample and various combinations of user-chosen values for the parameters. In essence, the parameters can be considered as factors and the values of the log-likelihood function, as responses. This set of data can then be stored directly on disk for input to the YATES program. Refer to section 3.3 for the methodology.

Figures 4.3.1 through 4.3.8 illustrate the procedure. First, a random sample of 100 values is generated from a Student's t distribution with parameters

$$\nu = 3, \quad \alpha = 5, \quad \sigma^2 = 9$$

Secondly, the log-likelihood function is evaluated using the sample and all possible combinations of the parameter

values

$\nu = 1, 4, 7, 10 \quad \alpha = 3, 5, 7 \quad \sigma^2 = 2, 10, 18$

The results are then placed on disk in YATES order (refer to section 3.3). In Figure 4.3.1, the user has already indicated to the program that the data are located on disk and has asked for a review of the input. Line 1 indicates that for the values

$\sigma^2 = 2 \qquad \alpha = 3 \qquad \nu = 1$

the value of the log-likelihood function is -341.6. Note that SIG SQ has 3 levels, ALPHA has 3 levels and NU has 4 levels. The user then proceeds with the analysis and produces the polynomial displayed in Figure 4.3.2. The error term should, of course, be disregarded. The coefficients of the polynomial appears to be somewhat difficult to comprehend; however, as it is merely an interpolating function, the user should not concern himself with it.

After several displays which allow the user to orient himself, he produces the contours given in Figure 4.3.3. Note that ALPHA is fixed at 6 and that a requested response of -292.00 is above the maximum of the surface for this particular combination of parameter values. The innermost contour suggests investigation of values of NU and SIG SQ of approximately 5.5 and 14, respectively. In Figure 4.3.4, this information has been used to fix SIG SQ at 14.0 and to plot NU against ALPHA. Contour 3, in turn, suggests investigation of values of NU and ALPHA of ap-

LIKELIHOOD FUNCTION EXAMPLE - FRAME 1

OUTPUT AREA

DEPRESS
KEY 1 TO CONTINUE WITH THE ANALYSIS
KEY 3 TO REPLACE DATA
DEPRESS KEY 28 TO PAGE BACKWARD.
DEPRESS KEY 29 TO PAGE FORWARD.

A = SIG SQ B = ALPHA C = NU

CELL	A	B	C	
1	1	1	1	-341.6
2	2	1	1	-323.1
3	3	1	1	-326.4
4	1	2	1	-313.2
5	2	2	1	-307.9
6	3	2	1	-315.5
7	1	3	1	-323.3
8	2	3	1	-314.4
9	3	3	1	-320.6
10	1	1	2	-394.8
11	2	1	2	-315.6
12	3	1	2	-310.0
13	1	2	2	-352.6
14	2	2	2	-298.6
15	3	2	2	-299.7
16	1	3	2	-370.1
17	2	3	2	-307.9
18	3	3	2	-306.2
19	1	1	3	-441.6
20	2	1	3	-321.9
21	3	1	3	-312.1
22	1	2	3	-390.3
23	2	2	3	-303.7
24	3	2	3	-300.6
25	1	3	3	-413.6
26	2	3	3	-314.4
27	3	3	3	-307.6
28	1	1	4	-477.7
29	2	1	4	-327.5
30	3	1	4	-314.0
31	1	2	4	-419.7
32	2	2	4	-308.5
33	3	2	4	-302.2
34	1	3	4	-447.8
35	2	3	4	-320.0
36	3	3	4	-309.5

REPLY AREA

FIGURE 4.3.1

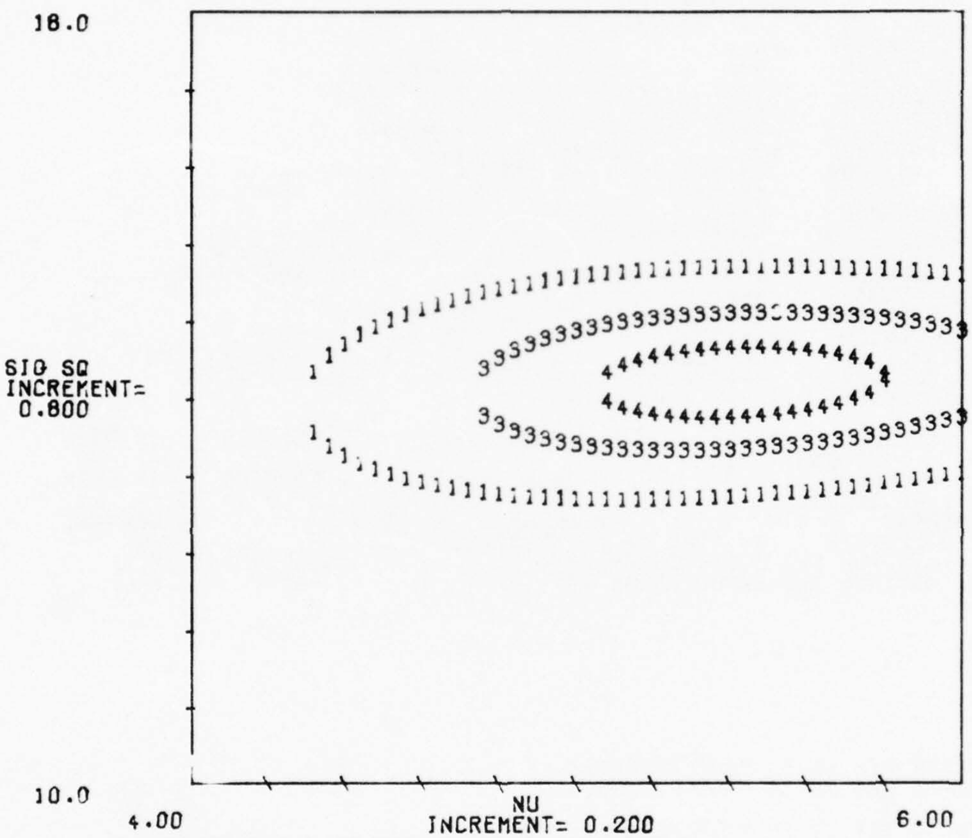
LIKELIHOOD FUNCTION EXAMPLE - FRAME 2

```
                         OUTPUT AREA
               FACTOR*A   IS  SIG SQ
               FACTOR*B   IS  ALPHA
               FACTOR*C   IS  NU

        IN STANDARD FACTOR VALUES. Y =
 -300.[illegible]554       +  36.50537     *A
 -35.26057     *A**2    +  3.780807     *B
 -5.561943     *A*B  +  3.997825     *A**2 *B
 -13.87952     *B**2    +  12.41254     *A*B**2
 -7.611929     *A**2 *B**2     -8.651396     *C
+  27.29918     *A*C    -20.97141     *A**2 *C
-0.1415557     *B*C    -1.256182     *A*B*C
+  1.317954     *A**2 *B*C    -1.926698     *B**2 *C
+  5.277975     *A*B**2 *C    -3.990466     *A**2 *B**2 *C
 -7.899030     *C**2     -7.632405     *A*C**2
+  5.851110     *A**2 *C**2    + 0.2693024     *B*C**2
+ 0.8519211     *A*B*C**2    -0.7351913     *A**2 *B*C**2
+ 0.8146361     *B**2 *C**2     -1.305243     *A*B**2 *C**2
+ 0.6695477     *A**2 *B**2 *C**2
+  8.312664     *C**3    +  2.577460     *A*C**3
 -2.046958     *A**2 *C**3    -0.1547698     *B*C**3
-0.3902204     *A*B*C**3    + 0.2706068     *A**2 *B*C**3
-0.2680662     *B**2 *C**3    + 0.3530042     *A*B**2 *C**3
-1.8850855E-01*A**2 *B**2 *C**3

   WHERE*A   = 0.0                * ARAW**2  + ( 0.1250000     )* ARAW
+ ( -1.250000      )
   WHERE*B   = 0.0                * BRAW**2  + ( 0.5000000     )* BRAW
+ ( -2.500000      )
 WHERE*C   = 0.0             * CRAW**3  + ( 0.0             )* CRAW**2
+(  0.2222222      )* CRAW  + ( -1.222221      )
   MINIMUM TWO SIGMA ERROR IS + OR -        2.02759  -=|xK|.
   MAXIMUM TWO SIGMA ERROR IS + OR -        2.84799  -=|xK|.


DEPRESS KEY 28 TO PAGE BACKWARD.
DEPRESS KEY 29 TO PAGE FORWARD.

DEPRESS KEY 30 TO VIEW OPTION TABLE
------------------------------------------------------------------
                         REPLY AREA
```

FIGURE 4.3.2

LIKELIHOOD FUNCTION EXAMPLE - FRAME 3

PLOT NO.	RESPONSE	ALPHA	
1.	-293.000	6.00000	
2.	-292.000	6.00000	NO SOLUTION
3.	-292.500	6.00000	
4.	-292.300	6.00000	

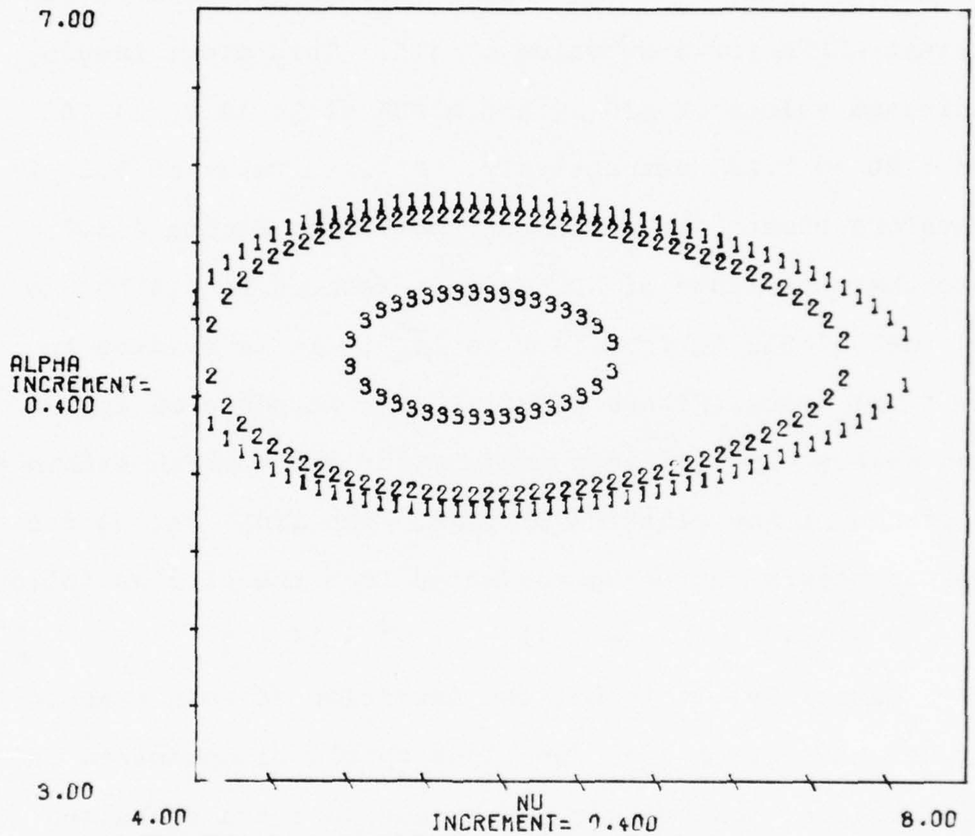


DEPRESS KEY 1 TO RETURN TO OPTION TABLE
OR ENTER NEW RESPONSE (NO MORE THEN FOUR MAY BE PLOTTED)

FIGURE 4.3.3

LIKELIHOOD FUNCTION EXAMPLE - FRAME 4

PLOT NO.	RESPONSE	SIG SQ	
1.	-292.300	14.0000	
2.	-292.000	14.0000	
3.	-291.000	14.0000	
4.	-290.500	14.0000	NO SOLUTION



DEPRESS KEY 1 TO RETURN TO OPTION TABLE
OR ENTER NEW RESPONSE (NO MORE THEN FOUR MAY BE PLOTTED)

FIGURE 4.3.4

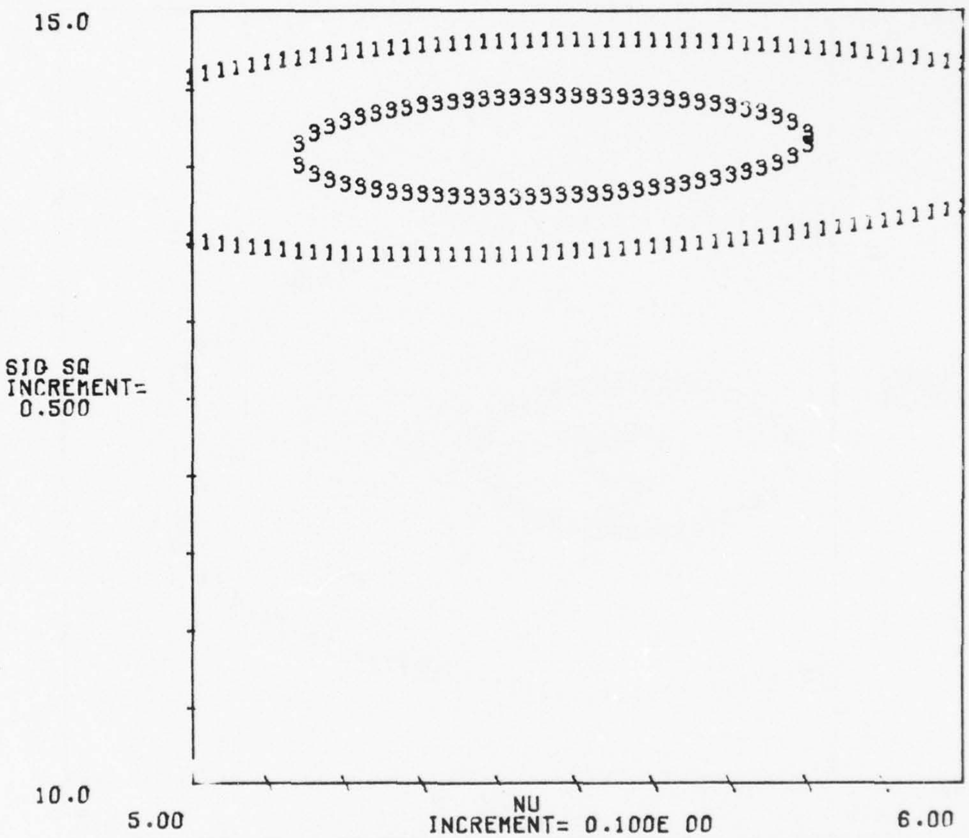
proximately 5.5 and 5.2, respectively. In Figure 4.3.5, ALPHA has been fixed at 5.2 and NU again plotted against SIG SQ. This plot leads the user to a value of 5.5 for NU and 14.0 to 14.5 for SIG SQ. Because NU has had a consistent value of about 5.5, Figure 4.3.6 displays SIG SQ against ALPHA for a NU value of 5.5. This plot, inturn, indicates values of SIG SQ and ALPHA of 14.10 to 14.15 and 5.20 to 5.22, respectively. A fixed value of 5.21 is therefore chosen for ALPHA thus producing Figure 4.3.7. Note that the range of NU has been reduced to 5.4 to 5.6 and that of SIG SQ from 14.1 to 14.2. It is evident in the final frame, Figure 4.3.8, that a response of approximately -290.743 is a maximum for the surface within the precision of the plotting program. The final values for the parameters can be approximated from the plot as follows:

$$\nu = 5.47 \qquad \alpha = 5.21 \qquad \sigma^2 = 14.125$$

Certainly, it is not the intention of this example to suggest that likelihood functions should be optimized in this fashion; however, it is obvious that the technique provides a valuable tool for the statistical researcher. First, the method may be used to provide good estimates of the parameters in order to initialize numerical optimization techniques. Secondly, the program may be used to verify that other procedures have in fact converged upon the absolute maximum of the function, not just a local one.

LIKELIHOOD FUNCTION EXAMPLE - FRAME 5

PLOT NO.	RESPONSE	ALPHA	
1.	-291.000	5.20000	
2.	-290.500	5.20000	NO SOLUTION
3.	-290.800	5.20000	
4.	-290.700	5.20000	NO SOLUTION

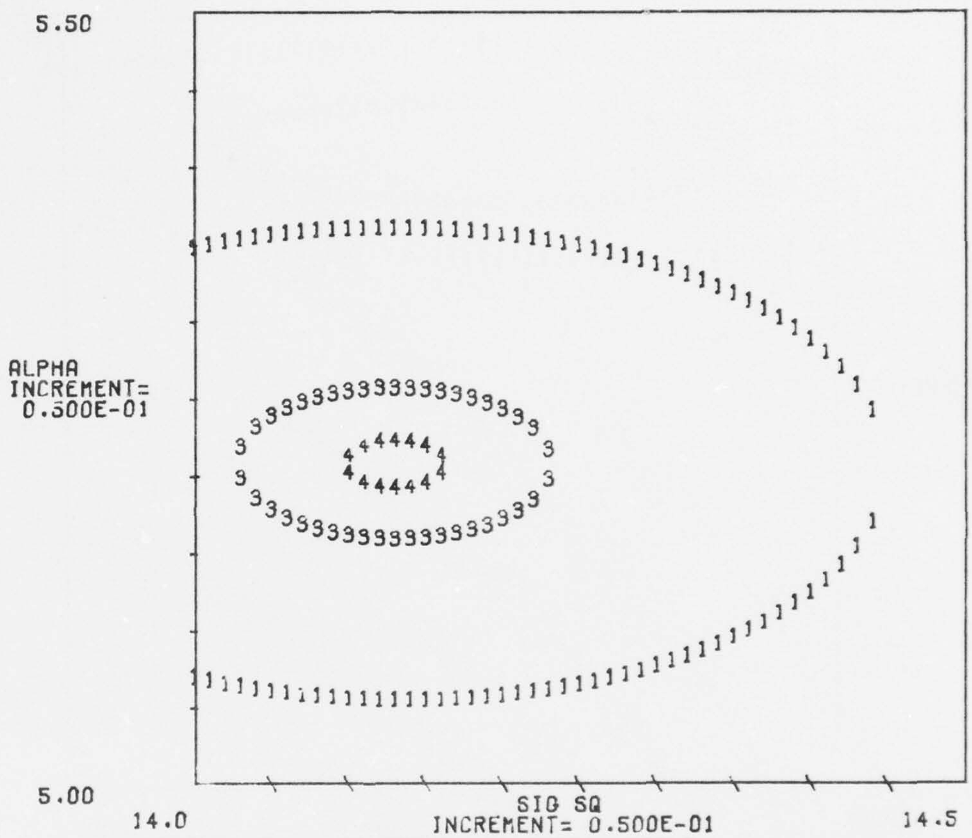


DEPRESS KEY 1 TO RETURN TO OPTION TABLE
OR ENTER NEW RESPONSE (NO MORE THEN FOUR MAY BE PLOTTED)

FIGURE 4.3.5

LIKELIHOOD FUNCTION EXAMPLE - FRAME 6

PLOT NO.	RESPONSE	NU	
1.	-290.800	5.50000	
2.	-290.700	5.50000	NO SOLUTION
3.	-290.750	5.50000	
4.	-290.745	5.50000	

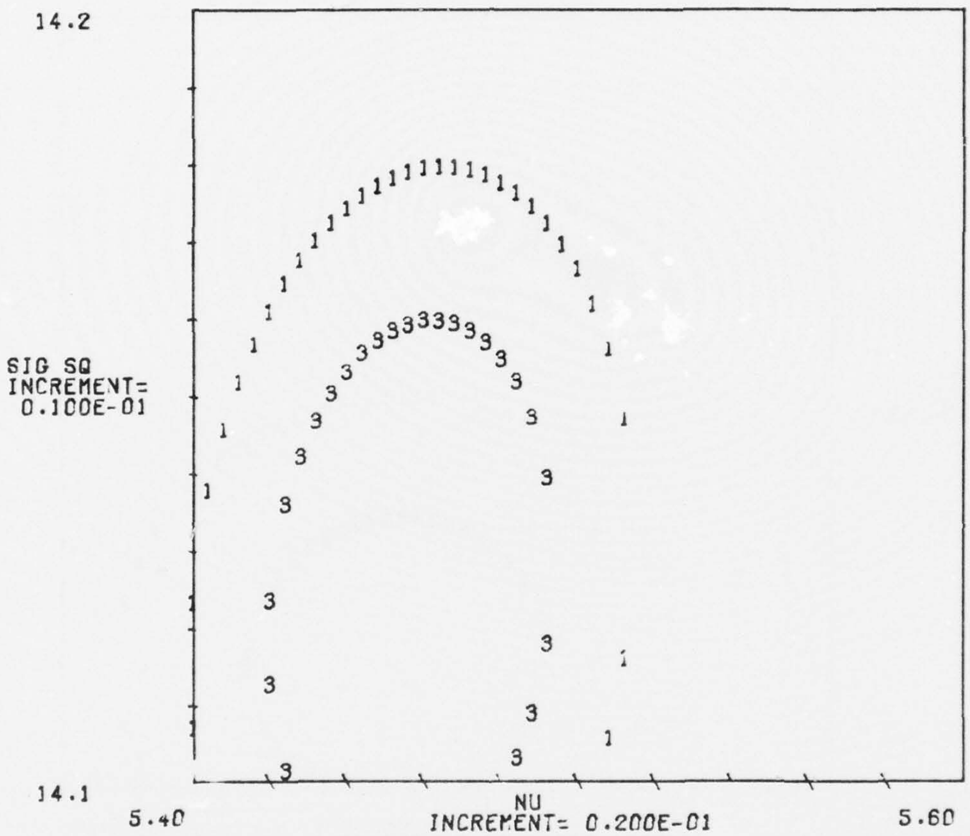


DEPRESS KEY 1 TO RETURN TO OPTION TABLE
OR ENTER NEW RESPONSE (NO MORE THEN FOUR MAY BE PLOTTED)

FIGURE 4.3.6

LIKELIHOOD FUNCTION EXAMPLE - FRAME 7

PLOT NO.	RESPONSE	ALPHA	
1.	-290.745	5.21000	
2.	-290.740	5.21000	NO SOLUTION
3.	-290.744	5.21000	
4.	-290.743	5.21000	NO SOLUTION

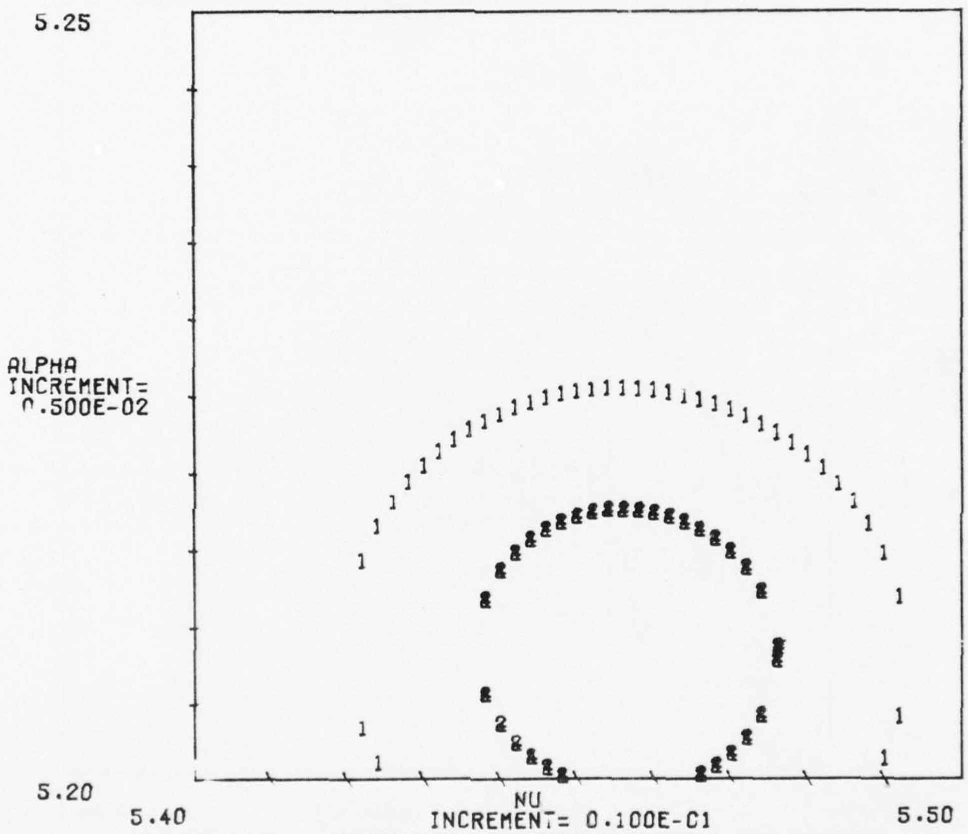


DEPRESS KEY 1 TO RETURN TO OPTION TABLE
OR ENTER NEW RESPONSE (NO MORE THEN FOUR MAY BE PLOTTED)

FIGURE 4.3.7

LIKELIHOOD FUNCTION EXAMPLE - FRAME 8

PLOT NO.	RESPONSE	SIG SQ	
1.	-290.744	14.1220	
2.	-290.743	14.1220	
3.	-290.743	14.1220	NO SOLUTION
4.	-290.743	14.1220	



DEPRESS KEY 1 TO RETURN TO OPTION TABLE
OR ENTER NEW RESPONSE (NO MORE THEN FOUR MAY BE PLOTTED)

FIGURE 4.3.8

CHAPTER V

APPLICATIONS OF GRAPHICS DISTRIBUTION UNIT

The statistician who has access to facilities which allow him to view immediate plots of statistical functions finds, as will be illustrated in this chapter, that the number of problem solving tools available to him has been significantly increased. Problems concerning the power of statistical tests which, in the past, have proved to be cumbersome and mathematically intractable become routine. In addition, determination of critical points for cumulative distribution functions whose inverses are difficult to calculate can be made accurately and inexpensively. The following sections serve to illustrate a few of the tasks to which the Graphics Distribution Unit may be applied.

5.1 The Power Function

To study the quality of the F-test of a general linear hypothesis, it is customary to plot or describe power functions for different levels of significance, or for different sample sizes. The power function plots power vs. values of the non-centrality parameter. In the F-test, such a plot requires redefinition of scaling of the axis

representing the non-centrality parameter. In terms of the non-central F distribution, as discussed in Bargmann [3], the non-centrality parameter is $\lambda = \underline{\gamma}'Q^{-1}\underline{\gamma}/\sigma^2$

where H_o: $C\ \underline{\Phi} = \underline{\gamma} = \underline{O}$

and a specified alternative is

$$H_a: C\ \underline{\Phi} = \underline{\gamma} \neq \underline{O} \text{ and } Q = C(X^1X)^{(-1)}C'$$

If sample size is fixed, λ itself can be used as the x axis. If, however, the power is to be compared for different sample sizes, it is preferable to use λ/N as the abscissa (Tang[24]). For example, consider the test of equality of means of a response under k different treatments:

$$H_o: \mu_1 = \mu_2 = \ldots = \mu_k$$

with r replications per treatment. For a given, specified alternative $(\mu_1, \mu_2, \ldots, \mu_k)$ the non-centrality parameter is

$$\lambda = r\sum_{i=1}^{k} (\mu_i - \mu)^2/\sigma^2$$

where

$$\lambda = (\mu_1+\mu_2+\ldots+\mu_k)/k$$

If we denote by δ^2 the average of the square of the deviation of treatment means from the grand mean, then

$$\lambda = rk\delta^2/\sigma^2 = N\delta^2/\sigma^2$$

Thus, λ/N is the square of the average deviation of treatment means from the grand mean, in standard units.

Example 5.1.1

Suppose we have k = 5 treatments and r = 7 observations per cell. Then

$m = k - 1 = 4$ (treatment degrees of freedom)

$n = k (r - 1) = 30$ (error degrees of freedom)

We wish to obtain the power function for values of α of .10, .05 and .01.

To obtain these diagrams we respond to the BLOWUP routine as follows:

Name of Parameter: ALPHA

Values of Parameter: .10, .05, .01

Minimum and Maximum for X: 0, 1.2

(this is λ/N in our notation)

Increment for X: .024 (51 points to be plotted)

Minimum and Maximum for Y: 0, 1

Function: Y = POWER(ALPHA,4,30,X)

The display is then in Figure 5.1.1. These plots are the desired power functions. To illustrate the blowup feature, let us assume that we wish to determine the noncentrality parameter for which the power is .90 under each of the stated significance levels. Viewing the plots we note that, somewhere between .4 and .6 there is the value

POWER FUNCTION EXAMPLE - FRAME 1

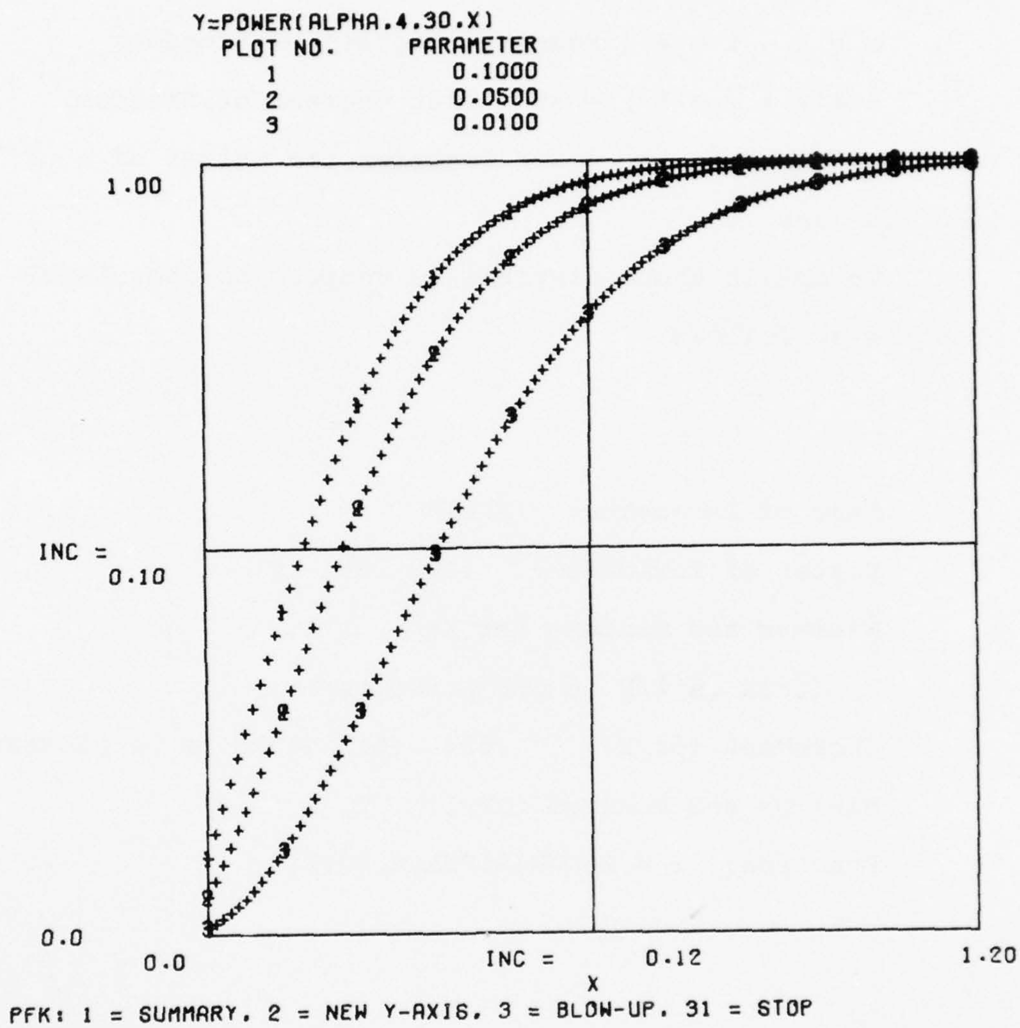


FIGURE 5.1.1

which has power of .90 for plots 1 (α=.10) and 2 (α=.05). The blowup, Figure 5.1.2, shows that λ/N = .42 for α=.10 and λ/N = .52 for α=.05; the power of .9 occurs for λ/N somewhere between .8 and .9 (Plot 3) if α=.01; the blowup Figure 5.1.3, shows λ/N = .74.

Thus the λ/N values for which a power of .90 is attained are .42, .52 and .74, respectively. To express these departures in terms of average standard deviations (δ/σ) we take square roots; thus the average standard deviations that produce a power of .90 and approximately, .65, .72 and .86 standard units for α=.10, .05 and .01, respectively.

Example 5.1.2

Suppose we have k = 4 treatments, and we wish to have a significance level of .01. What sample size do we need in order to attain a power of .90 for δ = .2σ, .3σ and .4σ? Note that the program is written for a general linear hypothesis, and thus operates in terms of degrees of freedom error. Recall

δ	.2σ	.3σ	.4σ
λ/N	.04	.09	.16

Responses to queries at the console are as follows:

POWER FUNCTION EXAMPLE - FRAME 2

OUTPUT AREA

Y=POWER(ALPHA.4.30.X)

PLOT NO.	PARAMETER
1	0.1000
2	0.0500
3	0.0100

0.40000	0.88613	0.80108	0.55716
0.42000	0.90083	0.82242	0.58733
0.44000	0.91384	0.84186	0.61638
0.46000	0.92530	0.85951	0.64422
0.48000	0.93539	0.87547	0.67078
0.50000	0.94423	0.88985	0.69599
0.52000	0.95196	0.90277	0.71984
0.54000	0.95868	0.91433	0.74235
0.56000	0.96451	0.92465	0.76354
0.58000	0.96955	0.93384	0.78341
0.60000	0.97391	0.94201	0.80199

DEPRESS PFK 1 FOR SUMMARY
2 TO REPLOT
3 FOR FURTHER BLOW-UP
31 TO STOP

REPLY AREA

FIGURE 5.1.2

POWER FUNCTION EXAMPLE - FRAME 3

OUTPUT AREA

```
Y=POWER(ALPHA.4.30.X)

  PLOT NO.      PARAMETER
     1             0.1000
     2             0.0500
     3             0.0100

     0.70000      0.98829      0.97077      0.87677
     0.71000      0.98920      0.97276      0.88274
     0.72000      0.99006      0.97462      0.88847
     0.73000      0.99084      0.97637      0.89397
     0.74000      0.99158      0.97800      0.89923
     0.75000      0.99225      0.97954      0.90426
     0.76000      0.99288      0.98097      0.90908
     0.77000      0.99345      0.98231      0.91368
     0.78000      0.99398      0.98355      0.91808
     0.79000      0.99447      0.98472      0.92228
     0.80000      0.99492      0.98580      0.92630

DEPRESS PFK  1 FOR SUMMARY
             2 TO REPLOT
             3 FOR FURTHER BLOW-UP
            31 TO STOP
```

REPLY AREA

FIGURE 5.1.3

Name of Parameter: LMDOVN

Values of Parameter: .04, .09, 16 (λ/N)

Minimum and Maximum for X: 2, 502 (d.f. error)

Increment for X: 5 (10 points)

Minimum and Maximum for Y: 0, 1 (power)

Function: Y = POWER(.01,3,X,LMDOVN)

The display is shown in Figure 5.1.4. Figure 5.1.5 is the blowup which straddles a power of .90 in Plot 3 (λ/N = .16, N between 50 and 150) which occurs at N(d.f. error) $\doteq$ 120 (sample size $\doteq$ 125) and Figure 5.1.6 shows the blowup for Plot 2 (λ/N = .09) where the power of .9 occurs at d.f. error $\doteq$ 215 (sample size $\doteq$ 220). Thus to attain the desired discriminatory power of .90 against an alternative which is an average of 0.4 standard units away from H_o, we need a sample size of 125, and for 0.3 standard units we need 220 (if α = .01). Because of the rather gradual slope of Plot 1 in Figure 5.1.1, an extension of the value of N was made in Figure 5.1.7 (d.f.error from 400 to 600, Y from .75 to 1.0). The blowup in Figure 5.1.8 indicates that 484 degrees of freedom for error will be necessary to provide a power of .9 at a specified alternative for

H_a: $\delta = .2\sigma$

where δ is the average deviation of a treatment mean from the grand mean. The suggested number of replications

POWER FUNCTION EXAMPLE - FRAME 4

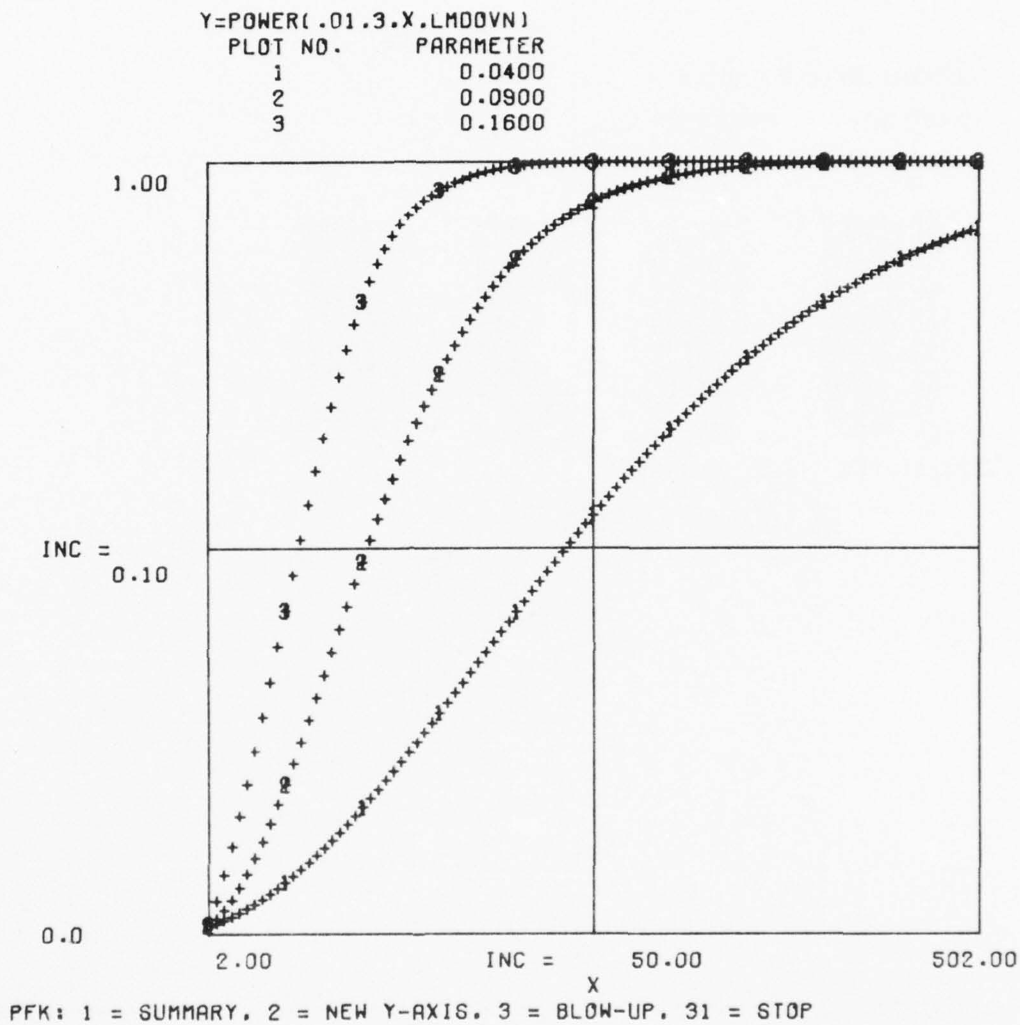


FIGURE 5.1.4

POWER FUNCTION EXAMPLE - FRAME 5

OUTPUT AREA

Y=POWER(.01.3.X.LMDOVN)

PLOT NO.	PARAMETER
1	0.0400
2	0.0900
3	0.1600

50.00000	0.06450	0.18356	0.40037
60.00000	0.08081	0.23754	0.49266
70.00000	0.09863	0.29465	0.58243
80.00000	0.11786	0.35348	0.66679
90.00000	0.13837	0.41264	0.74285
100.00000	0.16005	0.47075	0.80774
110.00000	0.18279	0.52650	0.85902
120.00000	0.20645	0.57929	0.89755
130.00000	0.23089	0.62882	0.92569
140.00000	0.25596	0.67480	0.94580
150.00000	0.28152	0.71694	0.96023

DEPRESS PFK 1 FOR SUMMARY
2 TO REPLOT
3 FOR FURTHER BLOW-UP
31 TO STOP

REPLY AREA

FIGURE 5.1.5

POWER FUNCTION EXAMPLE - FRAME 6

OUTPUT AREA

Y=POWER(.01.3.X.LMDOVN)

PLOT NO.	PARAMETER
1	0.0400
2	0.0900
3	0.1600

200.00000	0.41217	0.86949	0.99363
205.00000	0.42515	0.88005	0.99486
210.00000	0.43807	0.88988	0.99588
215.00000	0.45091	0.89902	0.99672
220.00000	0.46366	0.90749	0.99740
225.00000	0.47632	0.91534	0.99794
230.00000	0.48888	0.92260	0.99835
235.00000	0.50132	0.92930	0.99866
240.00000	0.51363	0.93548	0.99890
245.00000	0.52582	0.94118	0.99907
250.00000	0.53787	0.94643	0.99920

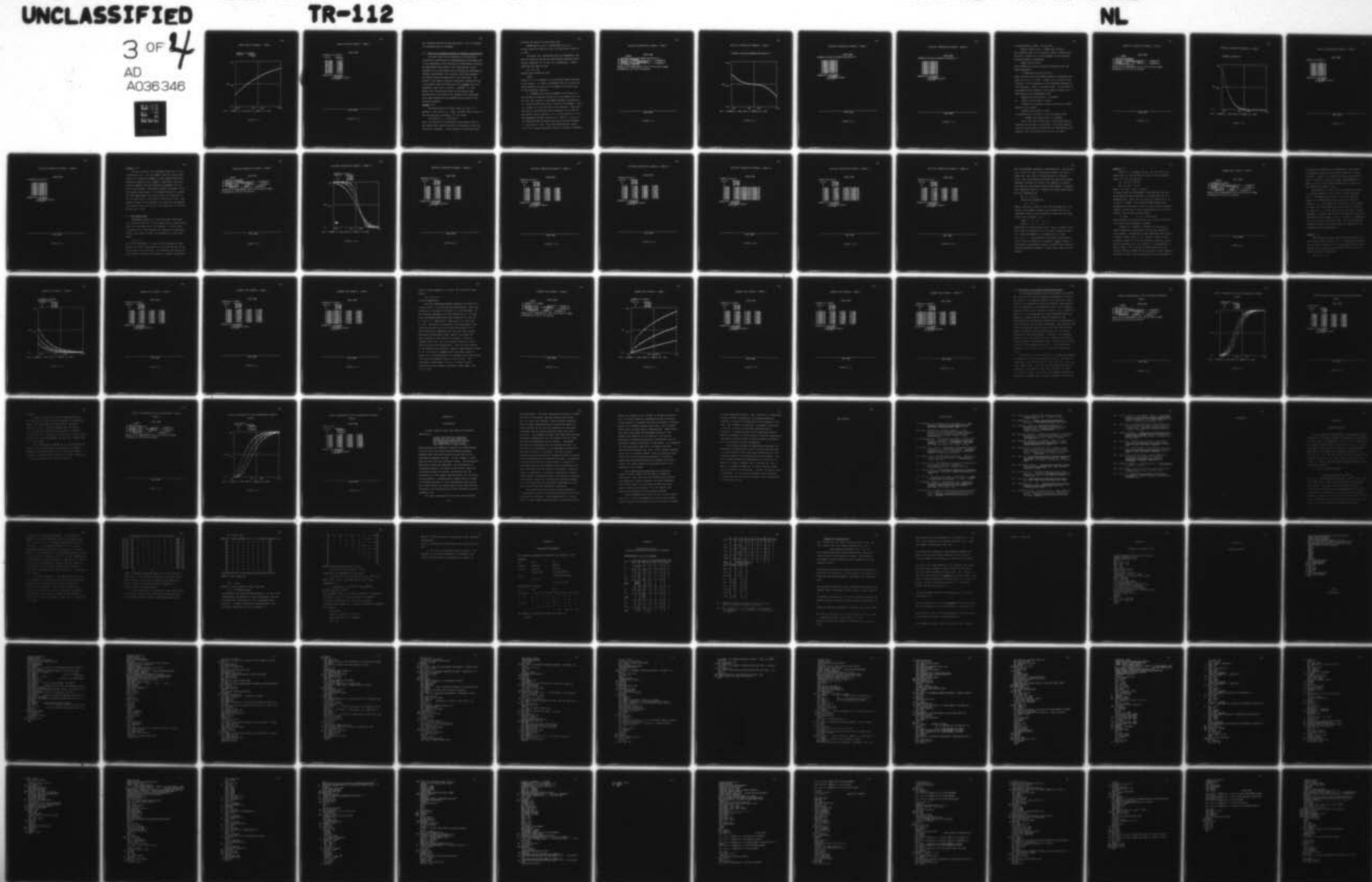
DEPRESS PFK 1 FOR SUMMARY
2 TO REPLOT
3 FOR FURTHER BLOW-UP
31 TO STOP

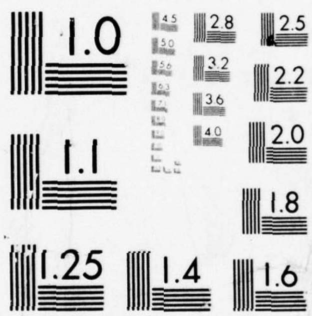
REPLY AREA

FIGURE 5.1.6

AD-A036 346 GEORGIA UNIV ATHENS DEPT OF STATISTICS AND COMPUTER--ETC F/G 9/2
GRAPHICAL AIDS FOR STATISTICAL COMPUTATION.(U)
DEC 76 W P BOND, R E BARGMANN N00014-69-A-0423
UNCLASSIFIED TR-112 NL

3 OF 4
AD
A036346




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6

POWER FUNCTION EXAMPLE - FRAME 7

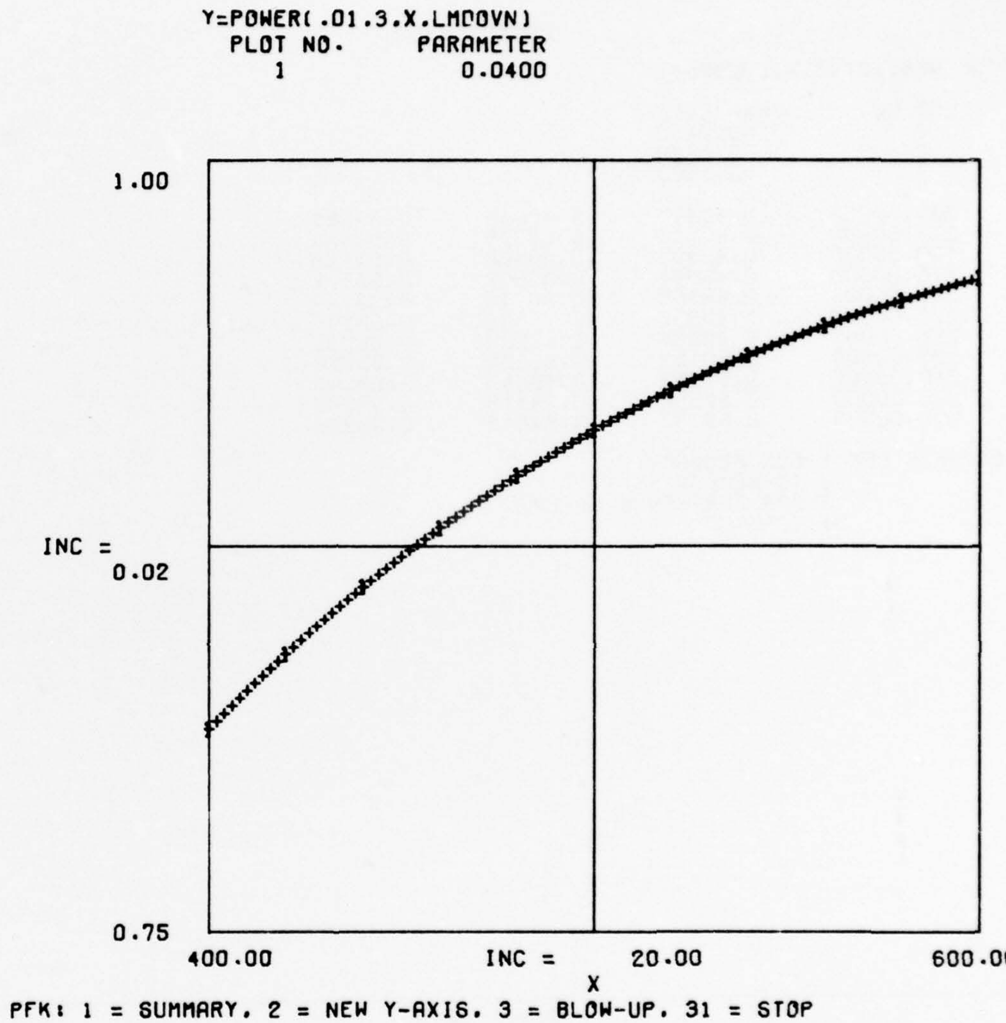


FIGURE 5.1.7

POWER FUNCTION EXAMPLE - FRAME 8

OUTPUT AREA

```
Y=POWER( .01.3.X.LMDOVN)

  PLOT NO.        PARAMETER
     1               0.0400

    480.00000       0.89775
    482.00000       0.89932
    484.00000       0.90086
    486.00000       0.90239
    488.00000       0.90390
    490.00000       0.90539
    492.00000       0.90685
    494.00000       0.90830
    496.00000       0.90973
    498.00000       0.91114
    500.00000       0.91253

DEPRESS PFK  1  FOR SUMMARY
             2  TO REPLOT
             3  FOR FURTHER BLOW-UP
            31  TO STOP
```

REPLY AREA

FIGURE 5.1.8

per treatment should be more than 484/4 = 121 or probably 125 observations per treatment.

5.2 Tests and Confidence Bounds for Multiple Correlations

An exact test for the equality of two multiple correlation coefficients is mathematically intractable due to the complexity of the statistical distribution involved. An approximate test based on the distribution of the quotient of two non-central Beta variates was developed by Schumann and Bradley [22]; however, this test assumes a non-central Beta distribution of the first kind. The square of the sample multiple correlation coefficient has a non-central Beta distribution of the <u>second</u> kind, and, therefore, this test is invalid. Bargmann [3] and Thomas [25] discuss both types of non-central Beta distributions, and develop the computational techniques which were employed by the NCBETA function used in the following example.

<u>Example</u> 5.2.1

The distribution of $R_1^2-R_2^2$, given $\rho_1^2 = \rho_2^2 = \rho^2$, depends on the value of ρ. Thus, one must find a value of the non-centrality parameter (ρ^2) for which

$$P[R^2<r_1^2|\rho^2] = 1 - P[R^2>r_2^2|\rho^2]$$

Thus, non-central R^2 distributions (non-central Beta of the second kind) must be plotted as functions of the non-centrality parameter. In the Graphics Distribution Unit

we would thus use ρ^2 as the X-axis and

$$NCBETA(n_1^2,m,n_1,X,2) + NCBETA(n_2^2,m,n_2,X,2)-1$$

as the Y-axis; we look for that X at which this function is zero.

Say that, for 3 predictors and two independent samples of sizes 255 and 240 we found sample multiple correlations (squared) of .25 and .49, respectively. Using these data we wish to test

$$H_o: \rho_1^2 = \rho_2^2$$

against the alternative that

$$\rho_1^2 \neq \rho_1^2$$

Figure 5.2.1 is a summary of the plotting orders directed to the system. In order to determine the X for which the above equality is true we will attempt to find the root of the following function.

Y = BETNC5(.49,3,240,X,2)+BETNC5(.25,3,255,X,2)-1

The minimum and maximum values for X are chosen to be .25 and .49, the squares of the sample multiple correlations. It is obvious that the value of Y will fall between -1 and +1. Figure 5.2.2 is the plot of the function. Note that the root of the function occurs slightly to the left of the center line of the plot, i.e., to the left of X = .37. The subsequent blowup, Figure 5.2.3, from X = .36 to X = .38 indicates that the zero value is to be found between X = .362 and X = .364. From the second blowup, Figure 5.2.4, it is clear that the X value (ρ^2 value) of interest

MULTIPLE CORRELATION EXAMPLE - FRAME 1

OUTPUT AREA

SUMMARY

1. Y=BETNC5(.49,3,240,X,2)+BETNC5(.25,3,255,X,2)-1
2. NO Z = G(X) SPECIFIED
3. MIN FOR X = 0.25000000 00 MAX FOR X = 0.49000000 00
4. INCREMENT FOR X = 0.20000000D-02
5. MIN FOR Y = -0.10000000 01 MAX FOR Y = 0.10000000 01
6. NO PARAMETER SPECIFIED.

DEPRESS THE KEY CORRESPONDING TO ANY ITEM YOU WISH TO CHANGE

OR DEPRESS KEY 28 TO PLOT THE FUNCTION

REPLY AREA

FIGURE 5.2.1

MULTIPLE CORRELATION EXAMPLE - FRAME 2

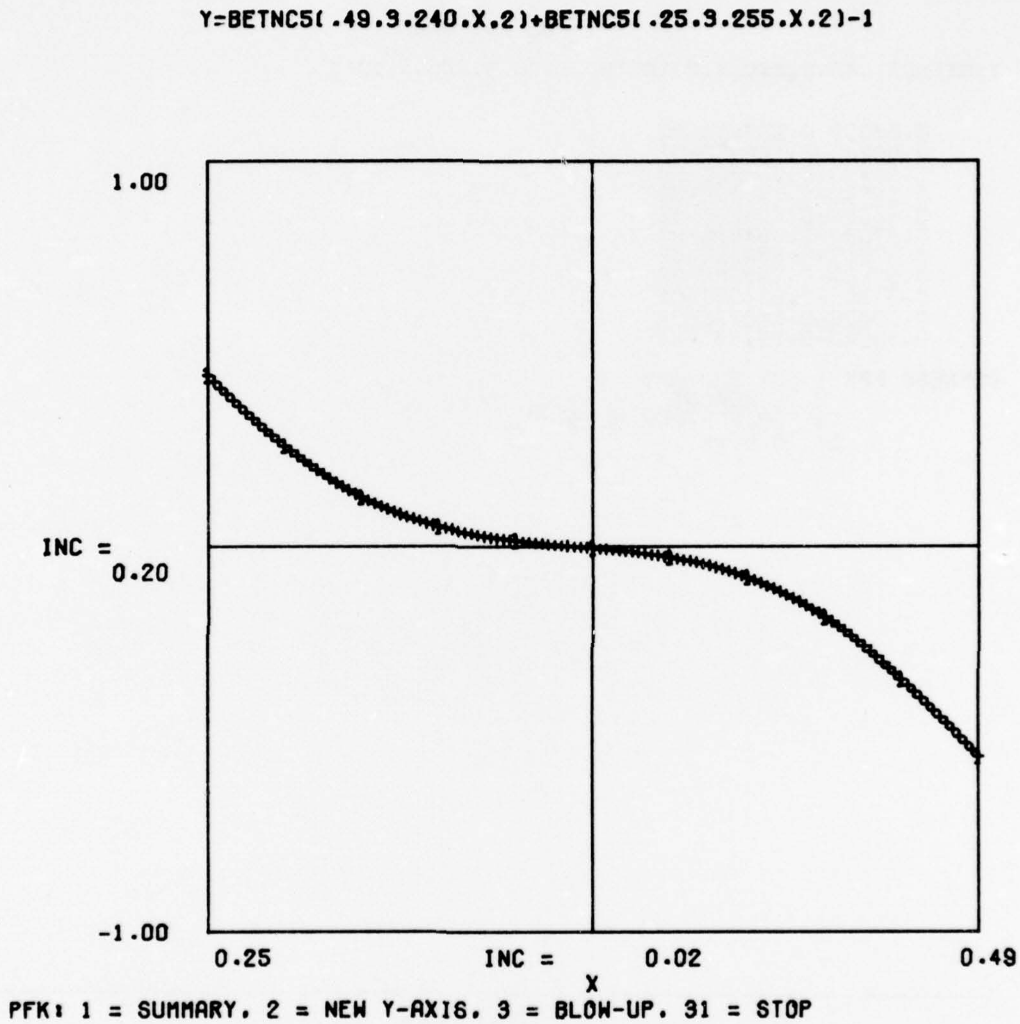


FIGURE 5.2.2

MULTIPLE CORRELATION EXAMPLE - FRAME 3

```
                              OUTPUT AREA
Y=BETNC5(.49.3.240.X.2)+BETNC5(.25.3.255.X.2)-1

    0.36000 0.25761E-02
    0.36200 0.10876E-02
    0.36400-0.38666E-03
    0.36600-0.18593E-02
    0.36800-0.33429E-02
    0.37000-0.48499E-02
    0.37200-0.69932E-02
    0.37400-0.79856E-02
    0.37600-0.96406E-02
    0.37800-0.11371E-01
    0.38000-0.13190E-01

DEPRESS PFK 1 FOR SUMMARY
            2 TO REPLOT
            3 FOR FURTHER BLOW-UP
           31 TO STOP
```

REPLY AREA

FIGURE 5.2.3

MULTIPLE CORRELATION EXAMPLE - FRAME 4

OUTPUT AREA

```
Y=BETNC5(.49.3.240.X.2)+BETNC5(.25.3.255.X.2)-1

     0.36200 0.10876E-02
     0.36220 0.93974E-03
     0.36240 0.79202E-03
     0.36260 0.64443E-03
     0.36280 0.49694E-03
     0.36300 0.34954E-03
     0.36320 0.20222E-03
     0.36340 0.54961E-04
     0.36360-0.92239E-04
     0.36380-0.23940E-03
     0.36400-0.38653E-03

DEPRESS PFK  1 FOR SUMMARY
             2 TO REPLOT
             9 FOR FURTHER BLOW-UP
            31 TO STOP
```

REPLY AREA

FIGURE 5.2.4

is approximately 0.3635. We see that

$$P[R_1^2<0.25|\rho^2=0.3635] = P[R_2^2>0.49|\rho^2=0.3635]$$

The remaining task is to find the values of these probabilities. Figure 5.2.5 is the summary of the plotting orders necessary to determine

$$P[R_1^2<0.25|\rho^2=0.3635]$$

This probability can be calculated directly using the CALCG expression

Y = BETNC5(0.25,3,255,0.3635,2)

Thus, we wish to use the blowup feature to determine the value of Y for X = 0.3635. Figure 5.2.6, a plot of the function, is not necessary for the procedure because the X of interest, 0.3635, is already known. It is simply an intermediate step leading to the blowup in Figure 5.2.7 from which we read immediately that

BETNC5(.25,3,255,0.3635) = 0.0064809

or $$P[R_1^2<0.25|\rho^2=0.3635] \doteq 0.0065$$

Although our procedure assures us that as a consequence of our actions

$$P[R_2^2>0.49|\rho^2=0.3635 \doteq 0.0065$$

we confirm this in Figure 5.2.8 as we observe that

1 - BETNC5(.49,3,240,0.3635) $\doteq$ 0.0064995

Due to the fact that we have a two-tailed test we calculate the P value as 2x0.0065 = 0.013 and conclude that the two multiple correlations are significantly different at the .05 level but not at the .01 level.

MULTIPLE CORRELATION EXAMPLE - FRAME 5

OUTPUT AREA

SUMMARY

1. Y=BETNC5(.25.3.255.X.2)
2. NO Z = G(X) SPECIFIED
3. MIN FOR X = 0.25000000D 00 MAX FOR X = 0.49000000D 00
4. INCREMENT FOR X = 0.20000000D-02
5. MIN FOR Y = 0.0 MAX FOR Y = 0.50000000D 00
6. NO PARAMETER SPECIFIED.

DEPRESS THE KEY CORRESPONDING TO ANY ITEM YOU WISH TO CHANGE

OR DEPRESS KEY 28 TO PLOT THE FUNCTION

REPLY AREA

FIGURE 5.2.5

MULTIPLE CORRELATION EXAMPLE - FRAME 6

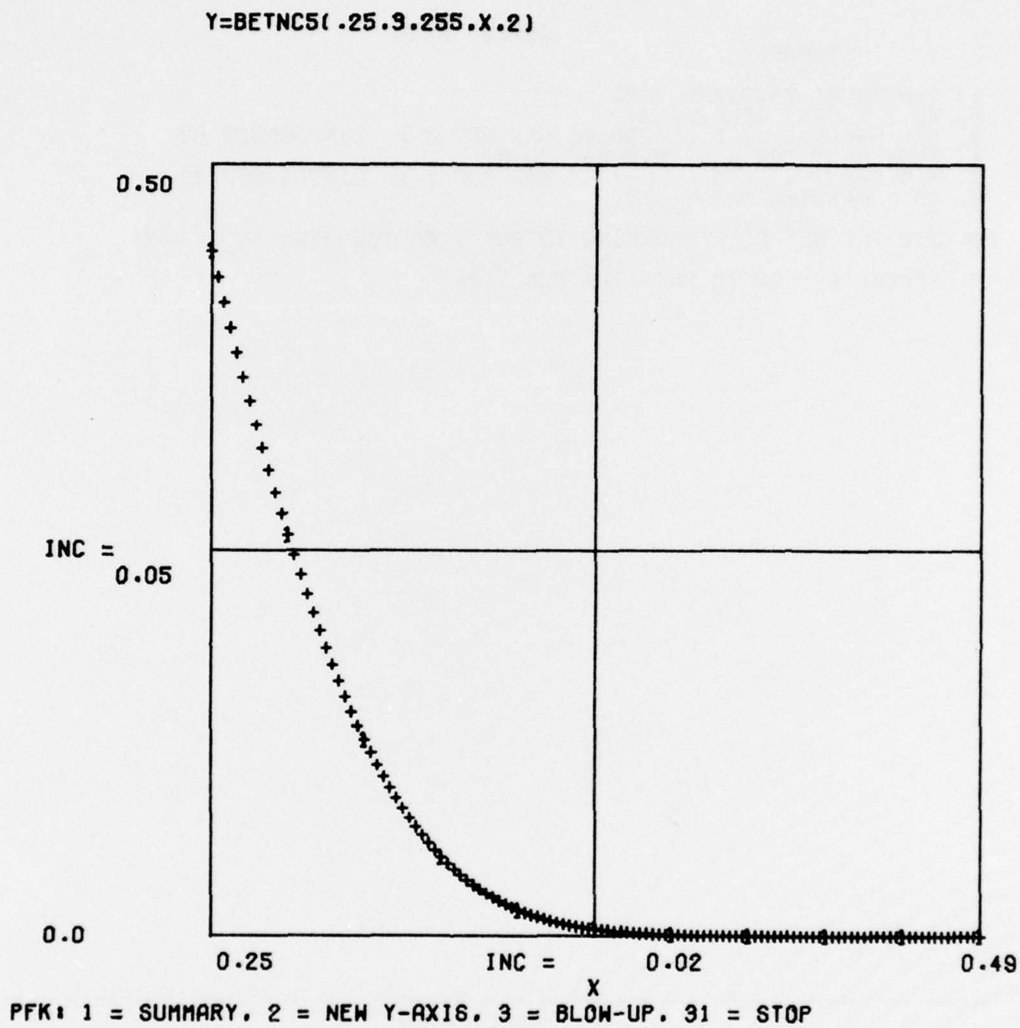


FIGURE 5.2.6

MULTIPLE CORRELATION EXAMPLE - FRAME 7

```
                    OUTPUT AREA
Y=BETNC5(.25.3.255.X.2)

     0.36300 0.66686E-02
     0.36310 0.66307E-02
     0.36320 0.65929E-02
     0.36330 0.65554E-02
     0.36340 0.65180E-02
     0.36350 0.64809E-02
     0.36360 0.64439E-02
     0.36370 0.64072E-02
     0.36380 0.63706E-02
     0.36390 0.63342E-02
     0.36400 0.62980E-02

DEPRESS PFK 1 FOR SUMMARY
            2 TO REPLOT
            3 FOR FURTHER BLOW-UP
           31 TO STOP
```

```
                    REPLY AREA
```

FIGURE 5.2.7

MULTIPLE CORRELATION EXAMPLE - FRAME 8

OUTPUT AREA

```
Y=1-BETNC5(.49.3.240.X.2)

     0.36300 0.63192E-02
     0.36310 0.63549E-02
     0.36320 0.63908E-02
     0.36330 0.64268E-02
     0.36340 0.64631E-02
     0.36350 0.64995E-02
     0.36360 0.65361E-02
     0.36370 0.65729E-02
     0.36380 0.66099E-02
     0.36390 0.66470E-02
     0.36400 0.66844E-02

DEPRESS PFK  1 FOR SUMMARY
             2 TO REPLOT
             3 FOR FURTHER BLOW-UP
            31 TO STOP
```

REPLY AREA

FIGURE 5.2.8

Example 5.2.2

We wish to obtain 95% confidence bounds on ρ^2 for 4 predictors, $r^2 = .49$, and sample sizes 50, 100 and 200. Figure 5.2.9 shows the summary of the commands which are especially simple in this example. Note that, as in the previous example, the non-centrality parameter (ρ^2) appears as the X-axis. The graph is shown in Figure 5.2.10. The blowups from Figure 5.2.11 through Figure 5.2.13 show that the approximate .025 value occurs at 0.2235 for N = 50, at 0.3180 for N = 100 and at 0.3765 for N = 200. Similarly, Figures 5.2.14 through 5.2.16 show the approximate .975 value to be 0.6435 for N = 50, 0.6065 for N = 100 and 0.5760 for N = 200.

5.3 The Lehmann Test

Hypothesis testing has frequently been criticized [15] on the ground that "if the sample size is large enough, every null hypothesis will be rejected." In fact, this criticism is not valid against the technique of hypothesis testing, but merely against the statement of a null hypothesis

$$H_0: \mu_1 = \mu_2$$

as a point hypothesis. If the two true population means differ by a very slight amount, H_0 will be rejected if the sample size is large enough. To illustrate the fallacy of the statement of the null hypothesis, consider the measure-

MULTIPLE CORRELATION EXAMPLE - FRAME 9

OUTPUT AREA

SUMMARY

```
1. Y=BETNC5(.49.4.N.X.2)
2. NO Z = G(X) SPECIFIED
3. MIN FOR X =  0.20000000D 00 MAX FOR X =  0.70000000D 00
4. INCREMENT FOR X =  0.50000000D-02
5. MIN FOR Y =  0.0            MAX FOR Y =  0.10000000D 01
6. THE VALUES FOR N           ARE:
     0.50000000D 02   0.10000000D 03   0.20000000D 03
DEPRESS THE KEY CORRESPONDING TO ANY ITEM YOU WISH TO CHANGE
OR DEPRESS KEY 28 TO PLOT THE FUNCTION
```

REPLY AREA

FIGURE 5.2.9

MULTIPLE CORRELATION EXAMPLE - FRAME 10

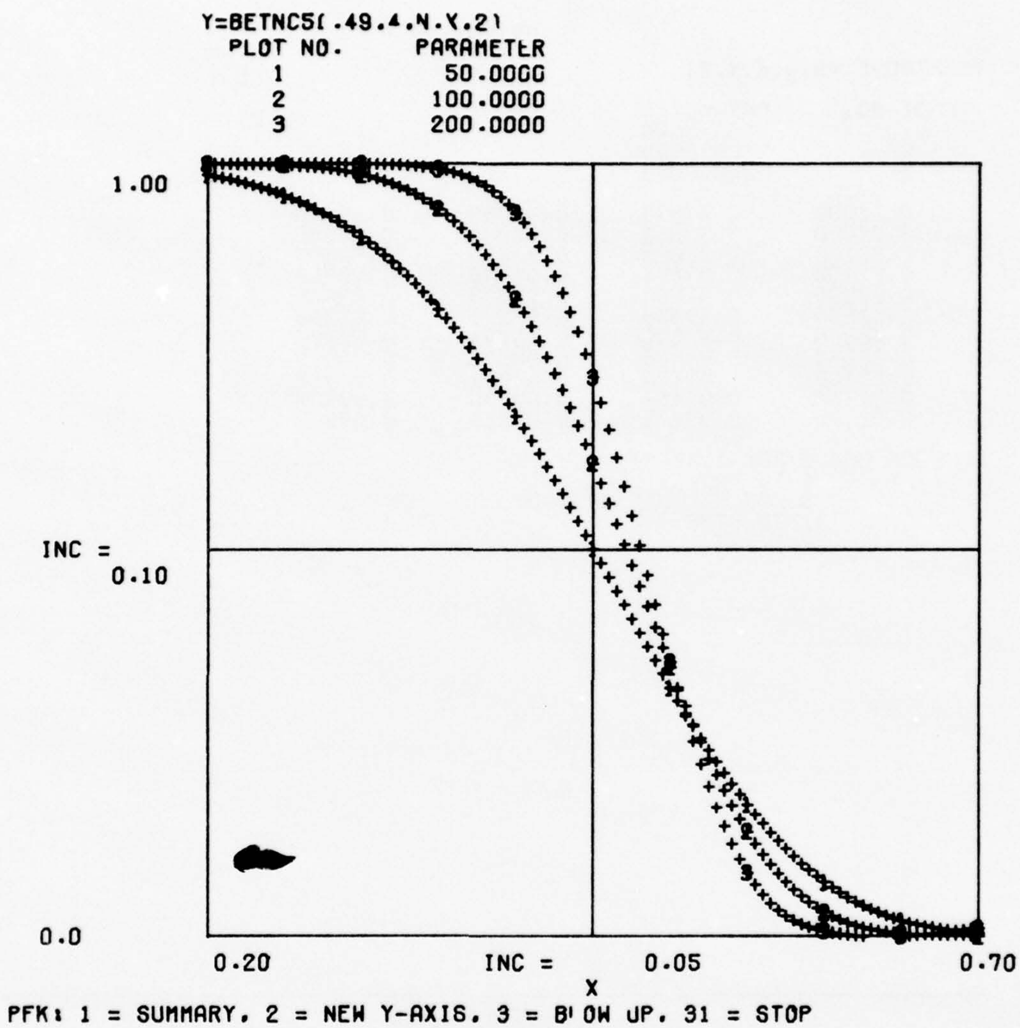


FIGURE 5.2.10

MULTIPLE CORRELATION EXAMPLE - FRAME 11

OUTPUT AREA

```
Y=BETNC5(.49.4.N.X.2)

  PLOT NO.       PARAMETER
     1             50.0000
     2            100.0000
     3            200.0000

     0.22000      0.97677     0.99883     0.99993
     0.22050      0.97653     0.99881     0.99993
     0.22100      0.97629     0.99880     0.99993
     0.22150      0.97604     0.99878     0.99993
     0.22200      0.97580     0.99877     0.99993
     0.22250      0.97555     0.99875     0.99993
     0.22300      0.97530     0.99874     0.99993
     0.22350      0.97505     0.99873     0.99993
     0.22400      0.97480     0.99871     0.99993
     0.22450      0.97454     0.99870     0.99993
     0.22500      0.97429     0.99868     0.99993

DEPRESS PFK 1 FOR SUMMARY
            2 TO REPLOT
            3 FOR FURTHER BLOW-UP
           31 TO STOP
```

REPLY AREA

FIGURE 5.2.11

MULTIPLE CORRELATION EXAMPLE - FRAME 12

OUTPUT AREA

Y=BETNC5(.49.4.N.X.2)

PLOT NO.	PARAMETER
1	50.0000
2	100.0000
3	200.0000

0.31500	0.88080	0.97721	0.99894
0.31550	0.88005	0.97687	0.99891
0.31600	0.87920	0.97653	0.99887
0.31650	0.87834	0.97618	0.99884
0.31700	0.87748	0.97583	0.99880
0.31750	0.87661	0.97548	0.99876
0.31800	0.87574	0.97512	0.99872
0.31850	0.87487	0.97476	0.99868
0.31900	0.87399	0.97439	0.99864
0.31950	0.87310	0.97402	0.99860
0.32000	0.87221	0.97364	0.99856

DEPRESS PFK 1 FOR SUMMARY
2 TO REPLOT
3 FOR FURTHER BLOW-UP
31 TO STOP

REPLY AREA

FIGURE 5.2.12

MULTIPLE CORRELATION EXAMPLE - FRAME 13

OUTPUT AREA

```
Y=BETNC5(.49.4.N.X.2)

  PLOT NO.      PARAMETER
     1            50.0000
     2           100.0000
     3           200.0000

     0.37500      0.74701      0.89392      0.97668
     0.37550      0.74562      0.89276      0.97617
     0.37600      0.74423      0.89159      0.97565
     0.37650      0.74283      0.89042      0.97512
     0.37700      0.74143      0.88923      0.97457
     0.37750      0.74003      0.88803      0.97402
     0.37800      0.73862      0.88683      0.97345
     0.37850      0.73721      0.88562      0.97288
     0.37900      0.73580      0.88439      0.97229
     0.37950      0.73438      0.88316      0.97169
     0.38000      0.73295      0.88192      0.97108

DEPRESS PFK 1 FOR SUMMARY
            2 TO REPLOT
            3 FOR FURTHER BLOW-UP
           31 TO STOP
```

REPLY AREA

FIGURE 5.2.13

MULTIPLE CORRELATION EXAMPLE - FRAME 14

OUTPUT AREA

```
Y=BETNC5(.49.4.N.X.2)

  PLOT NO.     PARAMETER
     1           50.0000
     2          100.0000
     3          200.0000

     0.64000      0.02734 0.57924E-02-0.52442E-03
     0.64050      0.02698 0.56034E-02-0.50958E-03
     0.64100      0.02662 0.54777E-02-0.49269E-03
     0.64150      0.02626 0.53554E-02-0.47376E-03
     0.64200      0.02591 0.52363E-02-0.45280E-03
     0.64250      0.02556 0.51206E-02-0.42983E-03
     0.64300      0.02521 0.50080E-02-0.40489E-03
     0.64350      0.02487 0.48986E-02-0.37798E-03
     0.64400      0.02453 0.47923E-02-0.34912E-03
     0.64450      0.02420 0.46891E-02-0.31834E-03
     0.64500      0.02386 0.45888E-02-0.28566E-03

DEPRESS PFK 1 FOR SUMMARY
            2 TO REPLOT
            3 FOR FURTHER BLOW-UP
           31 TO STOP
```

REPLY AREA

FIGURE 5.2.14

MULTIPLE CORRELATION EXAMPLE - FRAME 15

OUTPUT AREA

Y=BETNC5(.49.4.N.X.2)

PLOT NO.	PARAMETER
1	50.0000
2	100.0000
3	200.0000

0.60500	0.06324	0.02629	0.47445E-02
0.60550	0.06257	0.02580	0.45658E-02
0.60600	0.06190	0.02531	0.43905E-02
0.60650	0.06123	0.02483	0.42186E-02
0.60700	0.06057	0.02435	0.40501E-02
0.60750	0.05992	0.02389	0.38850E-02
0.60800	0.05927	0.02343	0.37233E-02
0.60850	0.05863	0.02297	0.35650E-02
0.60900	0.05799	0.02253	0.34099E-02
0.60950	0.05735	0.02209	0.32582E-02
0.61000	0.05672	0.02165	0.31098E-02

DEPRESS PFK 1 FOR SUMMARY
2 TO REPLOT
3 FOR FURTHER BLOW-UP
31 TO STOP

REPLY AREA

FIGURE 5.2.15

MULTIPLE CORRELATION EXAMPLE - FRAME 16

OUTPUT AREA

Y=BETNC5(.49.4.N.X.2)

PLOT NO.	PARAMETER
1	50.0000
2	100.0000
3	200.0000

0.57500	0.11360	0.07077	0.02656
0.57550	0.11260	0.06976	0.02591
0.57600	0.11161	0.06875	0.02526
0.57650	0.11062	0.06776	0.02463
0.57700	0.10964	0.06678	0.02401
0.57750	0.10866	0.06581	0.02341
0.57800	0.10768	0.06485	0.02282
0.57850	0.10672	0.06390	0.02224
0.57900	0.10576	0.06290	0.02168
0.57950	0.10480	0.06203	0.02112
0.58000	0.10385	0.06111	0.02058

DEPRESS PFK 1 FOR SUMMARY
2 TO REPLOT
3 FOR FURTHER BLOW-UP
31 TO STOP

REPLY AREA

FIGURE 5.2.16

ment of breakdown voltages of semiconductors. Values such as 3.5 volts (at most 2 significant digits) will be recorded for, say, each of 10,000 devices in 2 batches. Even if the two batches had exactly the same true average breakdown voltages per device, our inability to record more than two significant digits would produce a "significant" departure from H_o. Properly, we should have stated the null hypothesis as

$$H_o: |\mu_1 - \mu_2| \leq \delta$$

versus the alternative

$$|\mu_1 - \mu_2| > \delta$$

where δ must be at least 1/2 the last recorded unit (.05 volts in our example) Lehmann [12] proposes the use of "unbiased" tests for such hypothesis where the null hypothesis is an interval, i.e.,

$$\text{Power} > \alpha \quad \text{if} \quad |\mu_1 - \mu_2| > \delta$$

$$\text{Power} \leq \alpha \quad \text{if} \quad |\mu_1 - \mu_2| < \delta$$

determines the significance level. Thus, to obtain critical values for Lehmann's test, we need to know the value of t which corresponds to a probability of α when $|\mu_1 - \mu_2| = \delta$. This requires evaluation of non-central t (or t^2) with non-centrality parameter $|\frac{\mu_1-\mu_2}{\sigma}|$; tables of non-central distributions usually consider large values of the non-centrality parameter - here, small values are required.

<u>Example</u> 5.3.1

With $M = 2$ (3 groups) and $N_e = 20, 100, 200$ (i.e., sample size 23, 103 and 203) we wish to find critical F-values ($\alpha = .05$) for the test

$$H_o: |\mu_i - \mu|/\sigma \leq 0.125$$

$$\text{Alt}: |\mu_i - \mu|/\sigma > 0.125$$

where μ is $(n_1\mu_1 + n_2\mu_2 + n_3\mu_3)/n$

In other words, H_o is accepted if the deviation of each group mean from the grand mean is no more than 1/8 of a standard unit. Thus, the λ/N value (cf. section 5.1) is $(0.125)^2 = .015625$. The function POWER requires the probability of the Type I error (α) as the first argument, however, we want the F value corresponding to α as an abscissa. Thus we must use the option

Z = G(X)   Z = 1.0 - FFX 3(X,2,N)

where FFX 3(X,2,N) evaluates the c.d.f. of the F distribution with 2 and N degrees of freedom.

Figure 5.3.1 displays a summary of the plotting orders necessary to accomplish the task set forward by this example. The abscissa, which represents the critical F values, ranges from 2 to 10. These F values will be entered in steps of .08 to the secondary function (step 2) which calculates, as Z, the values of α corresponding to each F value. The Z values are then entered into the primary function, POWER, which calculates, as the ordinate, the power of the F test corresponding to the calculated α,

LEHMANN TEST EXAMPLE - FRAME 1

OUTPUT AREA

SUMMARY

```
1. Y=POWER(Z.2.N..015625)
2. Z=1.-FFX3(X.2.N)
3. MIN FOR X =   0.20000000D 01 MAX FOR X =  0.10000000D 02
4. INCREMENT FOR X =  0.80000000D-01
5. MIN FOR Y =  0.0           MAX FOR Y =  0.10000000D 01
6. THE VALUES FOR N          ARE:
     0.20000000D 02   0.10000000D 03   0.20000000D 03
```

DEPRESS THE KEY CORRESPONDING TO ANY ITEM YOU WISH TO CHANGE

OR DEPRESS KEY 28 TO PLOT THE FUNCTION

REPLY AREA

FIGURE 5.3.1

the degrees of freedom for treatments M=2, the degrees of freedom for error (N_e=20,100,200) and the λ/N value (0.15625). Figure 5.3.2 shows the display from which we wish to find the X values (F values) corresponding to a power of .05 (at the boundary between the ω_o region and the alternative region, $(\Omega-\omega_o)$.

The blowup of F from 4.0 to 5.0 is shown in Figure 5.3.3 which shows that the critical F value for N_e=20 is 4.1. The blowup of F from 5.0 to 6.0 is shown in Figure 5.3.4 which shows that the critical value for N_e=100 is 5.2. The blowup of F from 6.0 to 7.0 is shown in Figure 5.3.5 which shows that the critical F value for N_e = 200 is 6.7. We see then that the critical F values necessary to perform the Lehmann Test under the conditions specified in this example are 4.1, 5.2 and 6.7, respectively, for N values of 20,100 and 200. The F values may be compared with the values which would be required for the corresponding point hypothesis.

<u>Example</u> 5.3.2

The following example, dealing with the same situation investigated in example 5.3.1, is especially instructive in that it shows that, for large sample sizes, one is obliged to take more stringent significance levels, for testing the point hypothesis

$$H_o: \mu_1 = \mu_2 = \mu_3$$

LEHMANN TEST EXAMPLE - FRAME 2

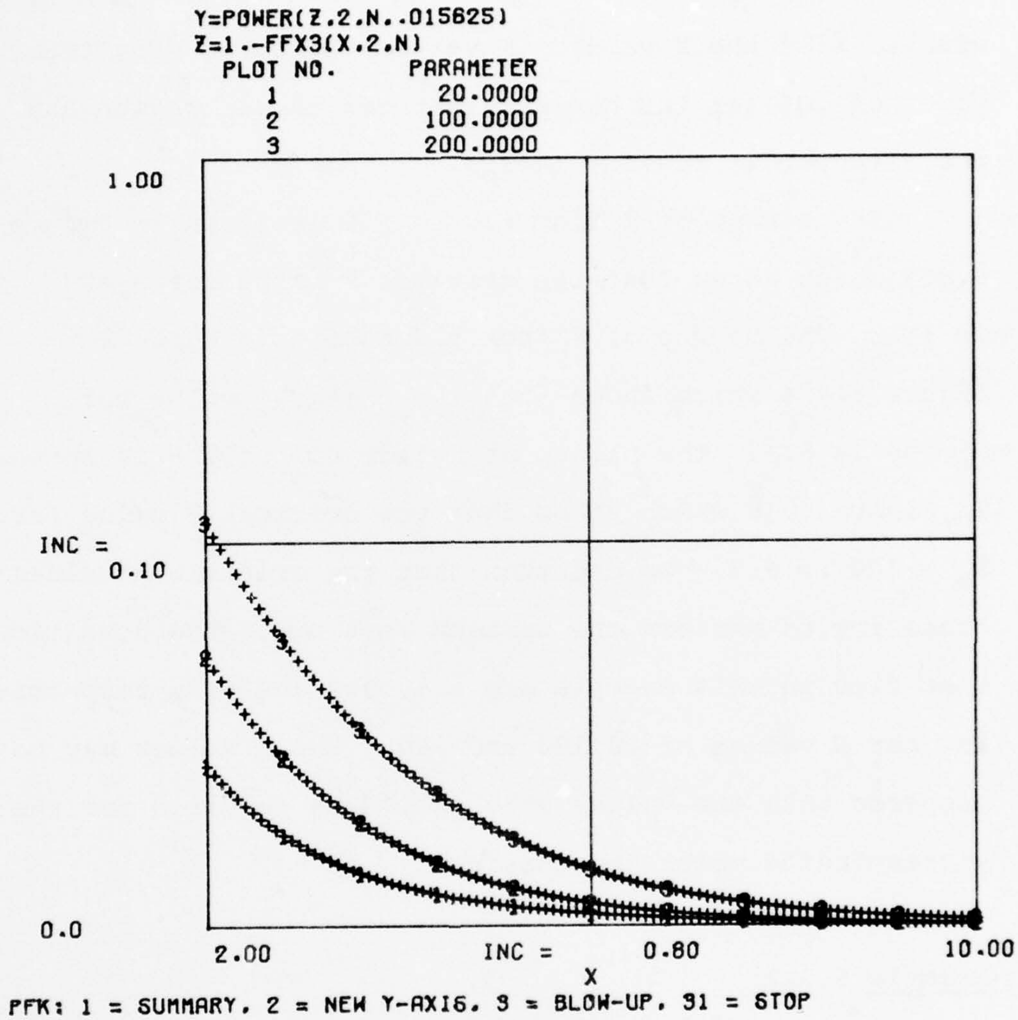


FIGURE 5.3.2

LEHMANN TEST EXAMPLE - FRAME 3

OUTPUT AREA

Y=POWER(Z.2.N..015625)

PLOT NO.	PARAMETER
1	20.0000
2	100.0000
3	200.0000

4.00000	0.05322	0.10493	0.21007
4.10000	0.04991	0.09858	0.19981
4.20000	0.04682	0.09260	0.18999
4.30000	0.04394	0.08697	0.18060
4.40000	0.04126	0.08168	0.17161
4.50000	0.03875	0.07669	0.16303
4.60000	0.03640	0.07200	0.15483
4.70000	0.03421	0.06759	0.14700
4.80000	0.03217	0.06344	0.13952
4.89999	0.03026	0.05954	0.13240
4.99999	0.02848	0.05588	0.12560

DEPRESS PFK 1 FOR SUMMARY
2 TO REPLOT
3 FOR FURTHER BLOW-UP
31 TO STOP

REPLY AREA

FIGURE 5.3.3

LEHMANN TEST EXAMPLE - FRAME 4

OUTPUT AREA

Y=POWER(Z.2.N..015625)

PLOT NO.	PARAMETER
1	20.0000
2	100.0000
3	200.0000

5.00000	0.02848	0.05588	0.12560
5.10000	0.02682	0.05244	0.11912
5.20000	0.02526	0.04921	0.11295
5.30000	0.02379	0.04617	0.10707
5.40000	0.02242	0.04331	0.10147
5.50000	0.02113	0.04063	0.09615
5.60000	0.01993	0.03811	0.09108
5.70000	0.01880	0.03575	0.08626
5.80000	0.01774	0.03353	0.08168
5.89999	0.01674	0.03145	0.07733
5.99999	0.01581	0.02949	0.07320

DEPRESS PFK 1 FOR SUMMARY
2 TO REPLOT
3 FOR FURTHER BLOW-UP
31 TO STOP

REPLY AREA

FIGURE 5.3.4

LEHMANN TEST EXAMPLE - FRAME 5

OUTPUT AREA

```
Y=POWER(Z.2.N..015625)

  PLOT NO.      PARAMETER
     1            20.0000
     2           100.0000
     3           200.0000

     6.00000 0.15807E-01      0.02949      0.07320
     6.10000 0.14929E-01      0.02766      0.06927
     6.20000 0.14103E-01      0.02599      0.06554
     6.30000 0.13327E-01      0.02431      0.06199
     6.40000 0.12599E-01      0.02280      0.05863
     6.50000 0.11914E-01      0.02137      0.05544
     6.60000 0.11270E-01      0.02004      0.05242
     6.70000 0.10665E-01      0.01879      0.04955
     6.80000 0.10095E-01      0.01761      0.04684
     6.89999 0.95573E-02      0.01651      0.04426
     6.99999 0.90508E-02      0.01548      0.04182

DEPRESS PFK 1 FOR SUMMARY
            2 TO REPLOT
            3 FOR FURTHER BLOW-UP
           31 TO STOP
```

REPLY AREA

FIGURE 5.3.5

than for small samples if, in fact, the "interval" hypothesis

$$H_o: |\mu_i - \mu|/\sigma \leq \delta$$

is more appropriate.

The only difference between Figures 5.3.6 and 5.3.1 is that the $Z = G(X)$ function has been removed. The interpretation of the plots in Figure 5.3.7 is as follows: If the interval hypothesis is to be tested for $\alpha = .05$, the point hypothesis would need to be tested at $\alpha \doteq .03$, for $N = 20$; at $\alpha \doteq .008$, for $N = 100$; and at $\alpha \doteq .003$, for $N = 200$. The actual α-values are .032 from Figure 5.3.8, .0075 from Figure 5.3.9, and .0026 from Figure 5.3.10. This illustration emphasizes the fact that the (usually arbitrary) significance levels ought to be chosen not only according to the nature of the data -- which is commonly done (.05, .01 in the Social Sciences, .001 in Quality Control and Engineering) -- but also with regard to the sample sizes involved. Where an experimenter intends to use .05 levels of significance with sample sizes of around 20, he should decide to go perhaps as far down as the .001 level with sample sizes of 1,000 or more. The (arbitrary) index number .05 at $N = 20$ has a closer similarity to the (equally arbitrary) index number .001 at $N = 1,000$.

LEHMANN TEST EXAMPLE - FRAME 6

OUTPUT AREA

SUMMARY

```
1. Y=POWER(Z.2.N..015625)
2. Z=X
3. MIN FOR X =  0.0             MAX FOR X =  0.10000000 00
4. INCREMENT FOR X =  0.10000000-02
5. MIN FOR Y =  0.0             MAX FOR Y =  0.50000000 00
6. THE VALUES FOR N          ARE:
      0.20000000 02   0.10000000 03   0.20000000 03
```

DEPRESS THE KEY CORRESPONDING TO ANY ITEM YOU WISH TO CHANGE

OR DEPRESS KEY 28 TO PLOT THE FUNCTION

REPLY AREA

FIGURE 5.3.6

LEHMANN TEST EXAMPLE - FRAME 7

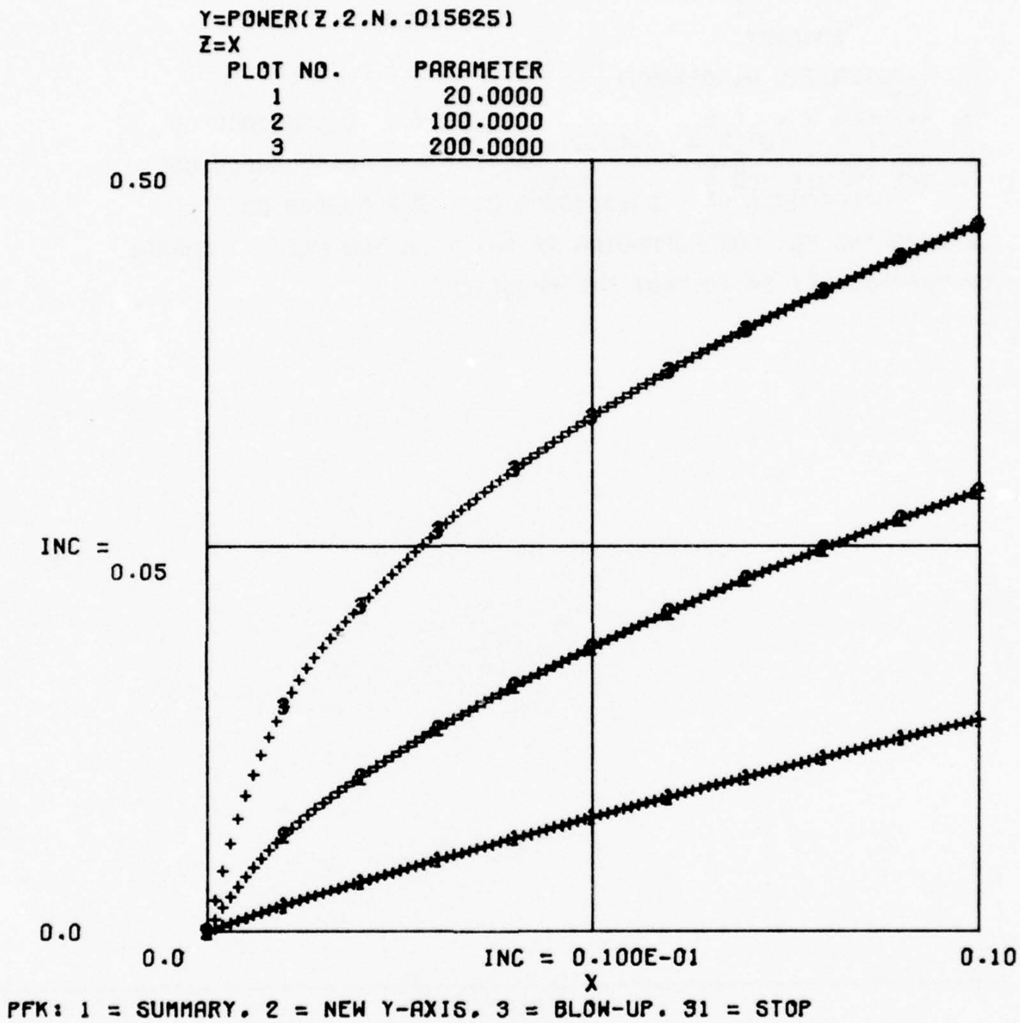


FIGURE 5.3.7

LEHMANN TEST EXAMPLE - FRAME 8

OUTPUT AREA

Y=POWER(Z.2.N..015625)

PLOT NO.	PARAMETER
1	20.0000
2	100.0000
3	200.0000

0.03000	0.04681	0.13183	0.25971
0.03200	0.04962	0.13762	0.26836
0.03400	0.05241	0.14324	0.27660
0.03600	0.05518	0.14872	0.28448
0.03800	0.05792	0.15407	0.29207
0.04000	0.06066	0.15930	0.29942
0.04200	0.06337	0.16444	0.30658
0.04400	0.06607	0.16949	0.31356
0.04600	0.06876	0.17445	0.32036
0.04800	0.07144	0.17933	0.32698
0.05000	0.07410	0.18412	0.33344

DEPRESS PFK 1 FOR SUMMARY
2 TO REPLOT
3 FOR FURTHER BLOW-UP
31 TO STOP

REPLY AREA

FIGURE 5.3.8

LEHMANN TEST EXAMPLE - FRAME 9

OUTPUT AREA

Y=POWER(Z.2.N..015625)

PLOT NO.	PARAMETER
1	20.0000
2	100.0000
3	200.0000

0.70000E-02	0.01235	0.04731	0.11430
0.71000E-02	0.01252	0.04785	0.11549
0.72000E-02	0.01268	0.04840	0.11668
0.73000E-02	0.01285	0.04894	0.11786
0.74000E-02	0.01302	0.04948	0.11902
0.75000E-02	0.01318	0.05002	0.12018
0.76000E-02	0.01335	0.05055	0.12132
0.77000E-02	0.01351	0.05108	0.12246
0.78000E-02	0.01368	0.05161	0.12359
0.79000E-02	0.01384	0.05213	0.12470
0.80000E-02	0.01400	0.05265	0.12581

DEPRESS PFK 1 FOR SUMMARY
2 TO REPLOT
3 FOR FURTHER BLOW-UP
31 TO STOP

REPLY AREA

FIGURE 5.3.9

LEHMANN TEST EXAMPLE - FRAME 10

OUTPUT AREA

Y=POWER(Z.2.N..015625)

PLOT NO.	PARAMETER
1	20.0000
2	100.0000
3	200.0000

0.20000E-02	0.36835E-02	0.01550	0.03946
0.21000E-02	0.38642E-02	0.01623	0.04128
0.22000E-02	0.40445E-02	0.01696	0.04308
0.23000E-02	0.42246E-02	0.01768	0.04487
0.24000E-02	0.44043E-02	0.01839	0.04664
0.25000E-02	0.45837E-02	0.01911	0.04840
0.26000E-02	0.47628E-02	0.01982	0.05015
0.27000E-02	0.49416E-02	0.02052	0.05188
0.28000E-02	0.51200E-02	0.02122	0.05360
0.29000E-02	0.52982E-02	0.02192	0.05530
0.30000E-02	0.54760E-02	0.02261	0.05700

DEPRESS PFK 1 FOR SUMMARY
2 TO REPLOT
3 FOR FURTHER BLOW-UP
31 TO STOP

REPLY AREA

FIGURE 5.3.10

5.4 Distribution of Largest Characteristic Root

The following example has been included to illustrate the incorporation into the Graphics Distribution Unit of a new function. As stated in section 3.4, there is a "short version" for inclusion of such new routines. The Function FUN1 (see Appendix D) evaluates the c.d.f. of the largest characteristic root of a matrix of rank 2, e.g., $T'(H+E)^{-1}T$ where $TT' = H$ for some general linear hypothesis with two degrees of freedom and (p) random variables, or the distribution of the canonical correlation, if the smaller of the two sets has two or fewer attributes. The explicit form of the distribution can be found in Roy [20] and involves several incomplete Beta-Functions which are, of course, contained in the CALCG package. As shown in the summary, (Figure 5.4.1) the function has two arguments in addition to the abscissa, representing m and n (or m_1 and m_2) in the notation of the Roy-Heck charts [10]. The illustration shows the c.d.f.'s for m = 8,9,10, and n=50 (see Figure 5.4.2).

The following blowup (Figure 5.4.3) shows the attempt to straddle the .99 percentage point for m=8, and the .95 percentage points for m=9 and 10. They are .313, .291, and .305, respectively. From the Roy-Heck charts we find for comparison .313 (Chart I, m=8, n=50) .29 and .305 (Chart III, m=9, 10; n=50). Of course, much greater precision is obtainable in BLOWUP, but is hardly necessary for practical

LARGEST CHARACTERISTIC ROOT DISTRIBUTION EXAMPLE

FRAME 1

```
                      OUTPUT AREA
      SUMMARY

 1. Y=FUN1(X,M,50)
 2. NO Z = G(X) SPECIFIED
 3. MIN FOR X =   0.10000000D 00 MAX FOR X =  0.40000000D 00
 4. INCREMENT FOR X =   0.30000000D-02
 5. MIN FOR Y =   0.0            MAX FOR Y =  0.10000000D 01
 6. THE VALUES FOR M          ARE:
       0.80000000D 01   0.90000000D 01   0.10000000D 02

DEPRESS THE KEY CORRESPONDING TO ANY ITEM YOU WISH TO CHANGE

OR DEPRESS KEY 28 TO PLOT THE FUNCTION
```

```
                      REPLY AREA
```

FIGURE 5.4.1

LARGEST CHARACTERISTIC ROOT DISTRIBUTION EXAMPLE

FRAME 2

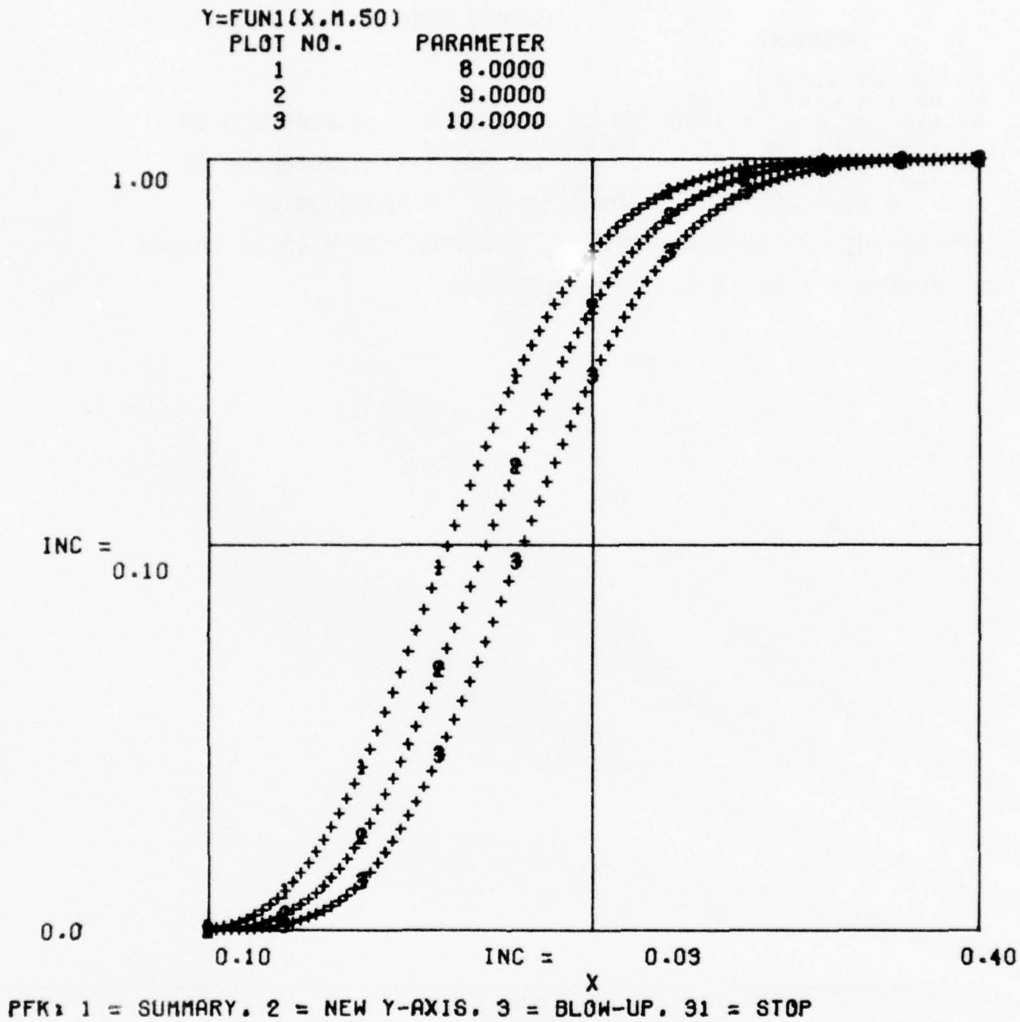


FIGURE 5.4.2

LARGEST CHARACTERISTIC ROOT DISTRIBUTION EXAMPLE

FRAME 3

OUTPUT AREA

Y=FUN1(X.M.50)

PLOT NO.	PARAMETER
1	8.0000
2	9.0000
3	10.0000

0.28000	0.95879	0.92648	0.87928
0.28600	0.96749	0.94075	0.90062
0.29200	0.97455	0.95262	0.91884
0.29800	0.98021	0.96239	0.93425
0.30400	0.98470	0.97035	0.94715
0.31000	0.98825	0.97680	0.95787
0.31600	0.99106	0.98202	0.96670
0.32200	0.99326	0.98617	0.97390
0.32800	0.99495	0.98944	0.97968
0.33400	0.99623	0.99198	0.98429
0.34000	0.99720	0.99395	0.98795

DEPRESS PFK 1 FOR SUMMARY
2 TO REPLOT
3 FOR FURTHER BLOW-UP
31 TO STOP

REPLY AREA

FIGURE 5.4.3

purposes.

Further, plots of the c.d.f's have been made for m= -1/2,0 and 1/2 (the latter not charted in the Roy-Heck charts, but attainable by interpolation), and n=20, (see Fig. 5.4.6). Again we straddle the .99 level in m = -1/2 and find .249 (.25 by Heck Chart I) and the .95 levels for m = 0 and m = 1/2, which are .22 and .253 in Figure 5.4.6 (.222 and between .222 and .280 in Heck Chart III, for n = 20).

These values are of importance to a statistician who wishes to place conservative (Roy-Scheffs) confidence bounds on individual comparisons in individual random variables in multivariate analysis-of-variance (outer box or polytope enclosing the elliptical region).

LARGEST CHARACTERISTIC ROOT DISTRIBUTION EXAMPLE

FRAME 4

OUTPUT AREA

SUMMARY

```
1. Y=FUN1(X,M,20)
2. NO Z = G(X) SPECIFIED
3. MIN FOR X =  0.0              MAX FOR X =  0.30000000 00
4. INCREMENT FOR X =  0.30000000D-02
5. MIN FOR Y =  0.0              MAX FOR Y =  0.10000000 01
6. THE VALUES FOR M         ARE:
     -0.50000000 00   0.0              0.50000000 00
```

DEPRESS THE KEY CORRESPONDING TO ANY ITEM YOU WISH TO CHANGE

OR DEPRESS KEY 28 TO PLOT THE FUNCTION

REPLY AREA

FIGURE 5.4.4

LARGEST CHARACTERISTIC ROOT DISTRIBUTION EXAMPLE

FRAME 5

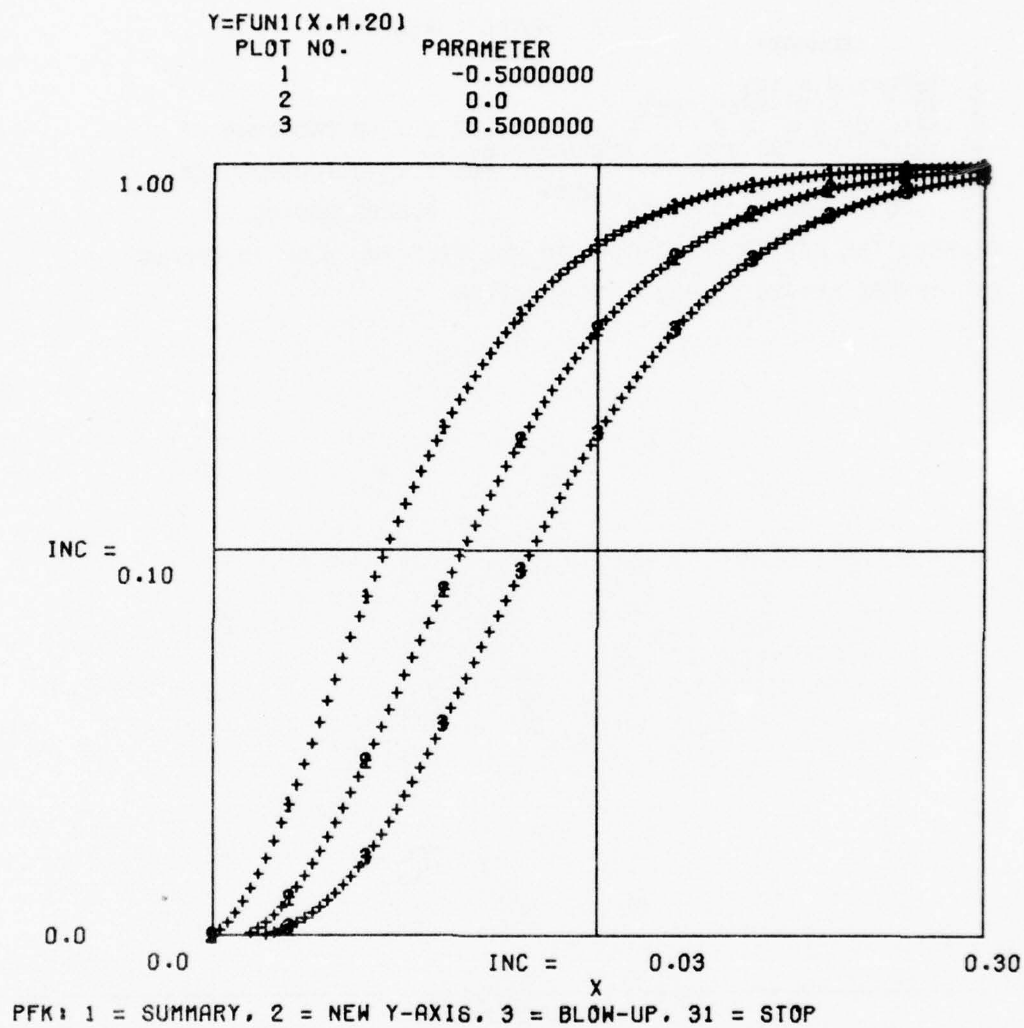


FIGURE 5.4.5

LARGEST CHARACTERISTIC ROOT DISTRIBUTION EXAMPLE

FRAME 6

OUTPUT AREA

Y=FUN1(X.M.20)

PLOT NO.	PARAMETER
1	-0.5000000
2	0.0
3	0.5000000

0.21000	0.97348	0.93601	0.87725
0.21600	0.97708	0.94393	0.89094
0.22200	0.98023	0.95096	0.90332
0.22800	0.98297	0.95720	0.91449
0.23400	0.98535	0.96271	0.92454
0.24000	0.98743	0.96758	0.93356
0.24600	0.98923	0.97187	0.94165
0.25200	0.99079	0.97565	0.94886
0.25800	0.99214	0.97896	0.95528
0.26400	0.99329	0.98184	0.96097
0.27000	0.99428	0.98435	0.96599

DEPRESS PFK 1 FOR SUMMARY
2 TO REPLOT
3 FOR FURTHER BLOW-UP
31 TO STOP

REPLY AREA

FIGURE 5.4.6

CHAPTER VI

CONCLUSIONS

In 1966, Tukey and Wilk [28] made the following observation:

> Current facilities for computing, display and real time interaction have developed substantially beyond our understanding of how to use them effectively in data analysis.

This research has attempted to expand our "understanding of how to use" such facilities by exploring several possible uses which the statistician may find for an interactive graphics terminal. In this respect, it has been involved with many different topics: the analysis of data from a factorial experiment, the optimization of likelihood surfaces, the power of the F-test, tests and confidence bounds for multiple correlations and the distribution of the largest characteristic root of certain matrix products. Although each of these areas is worthy of extended research, in this investigation these topics were chosen merely as illustrations to show the advantage provided to the statistician by having immediate access to graphical aids.

Two major computational tools have been developed

and implemented. The first concerns the fitting of tensor products of polynomials through rxsxtxu points where $2 \leq [r,s,t,u] \leq 4$, and $rstu \leq 256$. Such polynomial products were obtained, deterministically and statistically (by least squares) by the use of an extension of the Yates algorithm, and an inverse algorithm called the "Back Solution". The response surfaces thus generated have been represented graphically (for two factors with the other two used as parameters) and as contours. Confidence regions can be displayed for the statistical model, if the design is replicated, or an independent estimate of the error variance is available. The use of these facilities was illustrated for response surface estimation and the exploration of likelihood functions. An extension to five or more levels for each factor is relatively straightforward for the response surface estimation, but more complicated for the drawing of contours. Theoretically, an extension to orthogonal polynomials of higher degree poses no problems; however, the detailed implementation requires prepartion of extensive computer programs; it will be seen that the four-level algorithms are much more involved than the three-level approaches.

The second major tool relied on the existence of a comprehensive set of central and non-central statistical distribution functions. Since computation of a single point of some inverse distributions or non-central distri-

butions may require up to 100 MSEC of central processing time, and since graphical representation was the object of the research, a decision was made to evaluate a function exactly at 11 equally spaced points only, and to connect these points by cubic spline interpolation. These facilities were illustrated for hypothesis testing of the Lehmann type (where H_o is an interval), for testing equalities of multiple correlation coefficients, and for confidence interval estimation in the latter. To illustrate the incorporation of new units, an example has been included which displays the c.d.f. of the largest characteristic root of a matrix product (used in canonical correlation and in the multivariate analysis of variance). Graphically determined points were compared with Heck's charts [10], which served as verification both of our programs and the charts.

The programs were implemented on an IBM 2250 graphics console (1965 version) which is becoming obsolescent. The first task for further study would be the adaptation to newer equipment and other computers. Some initial work, on an Analysis-of-Covariance was done by Hayward and Bargmann [9], and should point the way for adaptation of the present programs.

Before considering the case of five or more levels per factor, it may be worthwhile to modify the extended Yates algorithm for handling fractional factorial designs,

or even undesigned studies. Also, especially in sequential response surface estimations, the program should be extended to store the information from one session to the next. The "BLOWUP" programs are, at present, restricted to the use of built-in functions, and functions of functions. To permit study of more complicated distributional problems, it may be useful to include interpretive facilities of the APL type, so that functions and mathematical operations can be concatenated.

The present programs are specifically designed for mathematical statisticians, thus no attempt has been made to include tutorial aids other than instructions for the technical aspects of conversation with the computer. The use of these programs for instruction is illustrated in section 5.3 where a Lehmann test is carried out, and in section 5.1 where the behavior of power functions under various degrees of alternatives, α levels, and sample sizes is displayed. In more modern equipment, such sequences of displays could be stored on magnetic disk and presented to students as a unit.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Ahlberg, J.H., Nilson, E.N., and Walsh, J.L. *The Theory of Splines and their Applications*, New York: Academic Press, 1967.

[2] Ball, Geoffrey H., and Hall, David J. "Some Implications of Interactive Graphic Computer Systems for Data Analysis and Statistics." *Technometrics*, 12, No. 1 (1970), 17-32.

[3] Bargmann, R.E. "Principles and Design of Statistical Computer Languages." *Progress in Operations Research*, III, ed. J.S. Aronofsky. New York: John Wiley and Sons, Inc., 1969, 293-316.

[4] Binet, F.E., et al. "Analysis of Confounded Factorial Experiments in Single Replication," *Technical Bulletin No. 11*, N.C. Agricultural Experiment Station. North Carolina, 1955.

[5] Bowman, K.O., and Kastenbaum, Marvin A. "Potential Pitfalls of Portable Power." *Technometrics*, 16 (1974), 349-352.

[6] Cooper, B.E. "The Extension of Yates' 2^n Algorithm to any Complete Factorial Experiment." *Technometrics*,10, No. 3 (1968), 575-577.

[7] Davies, Owen L. *The Design and Analysis of Industrial Experiments*, New York: Hafner Publishing Co., 1963.

[8] Fox, M. "Charts of the Power of the F-test." *Annals of Mathematical Statistics*, 29 (1956), 484-497.

[9] Hayward, Judith L., and Bargmann, R.E. *Methods of Conversion of Computer Dependent Interactive Programs*, THEMIS Report No. 30, Athens: University of Georgia, 1974.

[10] Heck, D.L. "Charts of Some Upper Percentage Points of the Distribution of the Largest Characteristic Root." *Annals of Mathematical Statistics*, 31. (1960), 625-642.

[11] Hunter, J.S. "The Inverse Yates Algorithm." Technometrics, 8, No. 1 (1966), 177-183.

[12] Lehmann, E.L. Testing Statistical Hypotheses, New York: John Wiley and Sons, Inc., 1959.

[13] Margolin, Barry H. "Systematic Methods for Analyzing $2^n 3^m$ Factorial Experiments with Applications." Technometrics, 9, No. 2 (1967), 245-260.

[14] Margolin, Barry H. "Design and Analysis of Factorial Experiments via Interactive Computing in APL." Technometrics, 18, No. 4 (1976), 135-150.

[15] Morrison, Denton E., and Henkel, Ramon E. (eds.) The Significance Test Controversy, Chicago: Aldine Publishing Co., 1970.

[16] Pearson, E.S., and Hartley, H.O. "Charts of the Power Function of the Analysis of Variance Tests, Derived From the Non-central F-Distribution." Biometrika, 38 (1951), 112-30.

[17] Penn, Lu. An On-line Statistical Computer System for Lay Usage, Technical Report No. 80, THEMIS Report No. 14, Athens: University of Georgia, 1971.

[18] Pennington, Ralph H. Introductory Computer Methods and Numerical Analysis, New York: The Macmillan Co., 1965.

[19] Poirier, D.J. "Piecewise Regression Using Cubic Splines." Journal of the American Statistical Association, 68, No. 343 (1973), 515-524.

[20] Roy, S.N. Some Aspects of Multivariate Analysis, New York: John Wiley and Sons, Inc., 1957.

[21] Schoenberg, I.J. (ed.) Approximations with Special Emphasis on Spline Functions, New York: Academic Press, 1969.

[22] Schumann, D.E.W., and Bradley, R.A. "The Comparison of the Sensitivities of Similar Experiments: Theory." Annals of Mathematical Statistics, 28 (1957), 902-920.

[23] Scott, Janice S., and Norman, James E. A Conversational Unit for Spline Function Construction, Technical Report No. 91, THEMIS Report No. 24, Athens: University of Georgia, 1972.

[24] Tang, P.C. "The Power Function of the Analysis of Variance Tests with Tables and Illustrations of Their Use." Statistic Research Memoirs, 2 (1938), 126-149.

[25] Thomas, Carlton G. Comparison of Non-central Beta Distribution Programs, Technical Report No. 62, THEMIS Report No. 10, Athens: University of Georgia, 1971.

[26] Tiku, M.L. "Tables of the Power of the F-test." Journal of the American Statistical Association, 62, No. 318 (1967), 525-539.

[27] Tiku, M.L. "More Tables of the Power of the F-test." Journal of the American Statistical Association, 67, No. 339 (1972), 709-710.

[28] Tukey, J.W., and Wilk, M.B. "Data Analysis and Statistics: An Expository Overview." Proceedings AFIPS Fall Joint Computer Conference, 29 (1966), 695-709.

[29] Wheeler, Robert E. "Portable Power." Technometrics, 16,(1974), 193-202.

[30] Wold, Svante "Spline Functions in Data Analysis." Technometrics,16, No. 1 (1974), 1-12.

[31] Yates, F. The Design and Analysis of Factorial Experiments. Imperial Bureau of Soil Science. Harpenden, 1937.

APPENDICES

APPENDIX A

ORTHOGONALIZATION

Let y_{ij} denote the observation on some output variable subjected to two factors (chosen for simplicity) each having 3 levels (again for simplicity). Let i = 1, 2, 3 denote the low, medium, and high level of factor A and j = 1, 2, 3 the low, medium, and high level of factor B. To avoid double subscripts, let A denote the z value for factor A, i.e., A=-1 for i=1, 0 for i=2, and 1 for i=3; and let B denote the z-value for factor B, i.e., B=-1 for j=1, 0 for j=2, and 1 for j=3. One can then write the "parametric" form of the 3^2 factorial model as

$$(1)\quad y_{ijk}=\mu+\alpha_1A+\alpha_2A^2+\beta_1B+\beta_2B^2$$

$$+\gamma_{11}AB+\gamma_{21}A^2B+\gamma_{12}AB^2+\gamma_{22}A^2B^2+e_{ijk}$$

where e_{ijk} is some experimental error. The α, β, and γ are the coefficients of the "response equation" for standard input values. The main use of this equation, once its coefficients have been estimated from experimental data, is for purposes of interpolation. By one of the conversion formulas in section 2.1, a pair of raw input values (T-values) can be translated into standard values(A is the z-value for factor B), and the y-value can be predicted (really,

interpolated in several dimensions). By statistical methods one can also determine which of the coefficients (the α's, β's, or γ's) contribute significantly to the prediction. If the combined effect of the α's is not appreciably greater than that of the e_{ijk}, we say that A effect is "non-significant", an analogous statement is made for the B effect in terms of the α's. If the γ effects are negligible we say that the model is "additive" (or, to use a more confusing statistical term, we say that there is "no significant interaction" between the two factors -- this latter statement should never be taken at its semantic face value!)

If further breakdown of contributions into "linear" and "quadratic" is desired we must first reformulated our polynomials in such a way that effect estimates are additive (uncorrelated). This can be attained by the classical technique of defining "orthogonal polynomials". This technique also permits solution of the estimation equations in simple, systematic, steps, and is illustrated as follows:

Disregarding errors we can write, explicitly

$$\begin{bmatrix} y_{11} \\ y_{21} \\ y_{31} \\ y_{12} \\ y_{22} \\ y_{32} \\ y_{13} \\ y_{23} \\ y_{33} \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \mu \\ \alpha_1 \\ \alpha_2 \\ \beta_1 \\ \beta_2 \\ \gamma_{11} \\ \gamma_{21} \\ \gamma_{12} \\ \gamma_{22} \end{bmatrix}$$

$$\underline{Y} = X\underline{\Phi}$$

(where it should be recalled that A=-1 if the first subscript is 1, 0 if the first subscript is 2, and +1 if the first subscript is 3; B is the same for the second subscript. This is the complete model of equation (1) spelled out for all observations.

If each observation is repeated (so that experimental error may be estimated) the best fitting values of $\underline{\Phi}$ can be estimated by the well-known least-squares equations.

(2) $X'X\underline{\Phi} = X'\underline{y}$

where X' is the transpose of X. In the above example X'X =

$$\begin{bmatrix} 9 & 0 & 6 & 0 & 6 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 \\ 6 & 0 & 6 & 0 & 4 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 6 & 0 & 0 & 4 & 0 & 0 \\ 6 & 0 & 4 & 0 & 6 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 & 4 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 & 0 & 0 & 4 & 0 \\ 4 & 0 & 4 & 0 & 4 & 0 & 0 & 0 & 4 \end{bmatrix}$$

Orthogonalization is achieved by translating the data matrix X into a matrix Z.

(3) $Z = XQ$

Where Q is to be chosen in such a way that

$$Z'Z = Q'X'XQ=D=\text{diagonal}$$

Q is chosen as an upper-triangular matrix. If (as in the illustration) the elements of X are polynomials, the set Q contains the coefficients of the "orthogonal polynomials". A simple Gram-Schmidt decomposition on the X'X matrix shows that Q, for our example, is

$$Q = \begin{bmatrix} 1 & 0 & -2/3 & 0 & -2/3 & 0 & 0 & 0 & 4/9 \\ & 1 & 0 & 0 & 0 & 0 & 0 & -2/3 & 0 \\ & & 1 & 0 & 0 & 0 & 0 & 0 & -2/3 \\ & & & 1 & 0 & 0 & -2/3 & 0 & 0 \\ & & & & 1 & 0 & 0 & 0 & -2/3 \\ & & & & & 1 & 0 & 0 & 0 \\ & & & & & & 1 & 0 & 0 \\ & & & & & & & 1 & 0 \\ & & & & & & & & 1 \end{bmatrix}$$

When we recall that the rows of X are

$1, A, A^2, B, B^2, AB, A^2B, AB\ \ , A^2B^2$

we see that the elements of Z=XQ will be

$1, A, A^2 - \frac{2}{3}, B, B^2 - \frac{2}{3}, AB, A^2B - \frac{2}{3}B, AB^2 - \frac{2}{3}.A,$

$A^2B^2 - \frac{2}{3}.B^2 - \frac{2}{3}.A^2 + \frac{4}{9}$ which can be written more compactly as

$1, A, \frac{1}{3}(3A^2-2), B, \frac{1}{3}(3B^2-2), AB, \frac{1}{3}(3A^2-2)B,$

$\frac{1}{3}(3B^2-2), \frac{1}{9}(3A^2-2)(3B^2-2)$

The expression $3A^2-2$ is called the quadratic "orthogonal polynomial" in A for 3 equally spaced (and equally weighted) discrete points. The usual algorithm multiplies the columns of Q in such a way that all elements are integers.

We now have

$X'X\underline{\hat{\Phi}} = X'\underline{y}$ (and $\underline{\hat{\Phi}}$ is desired)

XQ=Z such that Z'Z=D (diagonal)

$D\underline{\hat{\psi}} = Z'\underline{y}$

where $\underline{\hat{\psi}} = Q^{-1}\underline{\hat{\Phi}}$ are called "coefficients of the orthogonal polynomials";

hence $\underline{\hat{\psi}} = D^{-1}Z'\underline{y}$ are the solutions, and, given $\underline{\hat{\psi}}$ we can find

$\underline{\hat{\Phi}} = Q\underline{\hat{\psi}}$; this is called the "Back Solution". The elements of $Z'\underline{y}$ can be obtained in a systematic and simple procedure which is illustrated in Chapter II.

APPENDIX B

ORTHOGONAL POLYNOMIALS

The following Orthogonal Polynomials are assumed in this research

n=2	n=3	n=4
$\Phi_0(z)=1$	$\Phi_0(z)=1$	$\Phi_0(z)=1$
$\Phi_1(z)=z$	$\Phi_1(z)=z$	$\Phi_1(z)=3z$
	$\Phi_2(z)=3z^2-2$	$\Phi_2(z)=9/4z^2-5/4$
$z=1,1$		$\Phi_3(z)=45/4z^3-41/4z$
	$z=-1,0,1$	
		$z=-1,-1/3,1/3,1$

with values as follows

n=2

$\Phi_0(z)$	$\Phi_1(z)$
1	-1
1	1
(2)	(2)

n=3

z	$\Phi_0(z)$	$\Phi_1(z)$	$\Phi_2(z)$
-1	1	-1	1
0	1	0	-2
1	1	1	1
	(3)	(2)	(6)

n=4

z	$\Phi_0(z)$	$\Phi_1(z)$	$\Phi_2(z)$	$\Phi_3(z)$
-1	1	-3	1	-1
-1/3	1	-1	-1	3
1/3	1	1	-1	-3
1	1	3	1	1
	(4)	(20)	(4)	(20)

The number in parentheses below each column is

$$\sum_z \phi_i^2(z).$$

APPENDIX C

YATES METHOD FOR THE ANALYSIS OF BALANCED FACTORIAL EXPERIMENTS

Illustration: 3 x 2 x 3 design

	T	A	B	C	D_1	Cntr.	D_2	S. Squ.
1	18	57	96	180	18	10	1	1800
a	21	39	12	12	6	2	2	12
a^2	18	21	72	-144	6	-24	6	576
b	6	- 9	6	- 54	9	- 6	2	162
ab	21	39	- 4	18	3	6	4	27
a^2b	12	33	10	- 18	3	- 6	12	9
Sum_3		180		- 6				
c	5	0	-30	- 24	6	- 4	2	48
ac	14	6	-48	4	2	2	4	2
a^2c	2	- 3	-66	- 36	2	-18	12	54
Sum_2			48					
bc	- 7	- 1	-18	12	3	4	4	12
abc	6	0	- 30	4	1	4	8	2
a^2bc	- 8	10	- 6	24	1	24	24	24
Sum_3		12		- 16				

	T	A	B	C	D_1	Cntr.	D_2	S. Squ.
c^2	7	-6	6	144	6	24	6	576
ac^2	25	- 24	2	24	2	12	12	24
a^2c^2	7	- 21	10	0	2	0	36	0
bc^2	1	- 27	-18	36	3	12	12	36
abc^2	21	- 36	- 6	12	3	12	24	6
a^2bc^2	11	- 30	6	0	1	0	72	0
Sum_{23}		-144	-54	216				
Sum	180	48	- 6					
Even		- 3						
Odd		51						
Ev-Od		- 54						
Hi	42		26					
Mid	108		-74					
Lo	30		42					
Hi-Lo	12		-16					
Hi+Lo								
-2Hi	-144		216					

D_1 = (Number of times letter(s) in front occur as a factor) x(Numer of elements per cell)

D_2 = xyz , where x,y,z = 1 for absent, 2 for linear, 6 for quadratic, (= sum of squares of coefficients in contrast)

Comments for Illustration

For 2 Levels (B), the (only) linear contrast is $B_1 - B_0$

For 3 Levels (A,C) the linear contrast is $A_2 - A_0$

the quadratic contrast is $A_2 + A_0 - 2A_1$

The columns from which 3-level factors are to be calculated (T and B) are grouped in triples. The column(s) from which 2-level factors are to be calculated (A) are grouped in pairs.

The first six entries in A are the sums of the triples in T
The first six entries in C are the sums of the triples in B
These six entries are summed in the Check-Sum_3 (following a^2b)

The second six entries in A are the differences between the highest (last) and lowest (first) entries in each triple of T.

The second six entries in C are the differences between the highest (last) and lowest (first) entries in each triple of B.

These six entries are summed in the Check-Sum_3 (foll. a^2bc)

The last six entries in A (C) are obtained as first + last - 2 (middle) entries in each triple of T (B).

These six entries are summed in the Check-Sum_{23} at end of table.

The entries in B are obtained as in illustration 1. The first nine entries are the sums of each pair in A. They are summed in Check-Sum_2, foll. a^2c.

The second nine entries are the difference between the high (last) and low (first) entries in each pair of A. They are summed in Check-Sum_{23}, at the end of the table.

For T and B one forms the sum of all elements, the sum of all high (last) elements in each triple, and the sum of all low (first) elements of each triple. The columns T and B are those which are followed by 3-level factors. The Sum of column T agrees with the first Sum_3 of A. Hi-Lo of column T agrees with the second Sum_3 of A. Hi+Lo-2Mi agrees with the final Sum_{23} of A.

The same agreement holds for the table sums in B and the check-sums in C.

The table sums in A (a column followed by a 2-level factor) and the check-sums in B are the same as in illustration 1.

The elements of Cntr. (contrast estimates) are the entries of the (final) column C, each divided by D_1.

The elements of S.Squ. (sum of squares for each contrast)

equal C x Cntr./D_2.

APPENDIX D

FORTRAN IV LISTING OF FUN1

```
      DOUBLE PRECISION FUNCTION FUN1(X,AM,AN)
      IMPLICIT REAL*8(A,B,D,F)
      DOUBLE PRECISION X,COE
      DX = X
      AMP = AM + 1.DO
      ANP = AN + 1.DO
      DE = 1.DO
      FUN1 = DE
      IF(X.EQ.DE) RETURN
      FUN1 = O.DO
      IF(X.EQ.O.DO) RETURN
      IF(X*(DE-X) .LT.O.DO) RETURN
      IF(AMP.LE.O.1 .OR. ANP.LE.O1) RETURN
      AMPP = AMP+AMP
      ANPP = ANP+ANP
      A = BETAX(X,AMPP,ANPP)
      COE = DLGGM(AMPP+ANPP+DE)
     1 -DLGGM(AMPP+DE) - DLGGM(ANPP+DE)
      COE = COE - DLGGM(AMP+ANP+DE)
     1 +DLGGM(AMP+DE) + DLGGM(ANP+DE)
      COE = COE + AMP*DLOG(X) + ANP*DLOG(DE-DX)
      BB=BETAX(X,AMP,ANP)
      IF(BB.GT.O.DO) GO TO 14
      FUN1=A
      RETURN
   14 COE = COE + DLOG(BB)
      FUN1 = A - DEXP(COE)
      RETURN
      END
```

APPENDIX E

PROGRAM LISTINGS

```
      DEFINE FILE 18(250,80,E,IREC),19(270,6,U,JPOINT)
      DOUBLE PRECISION DSSQ,DER
      COMMON /YATDAT/ DSSQ,DER,
     3LVL(4),NFG,IRAW,DAT(256),NEND(4,4),TTL(20),
     1VNAM(2),NAME(8),FMT(80),ICODE,IL(6),NIN,NOUT,NDX,NF,
     2NVBL,NREP,CONCOM(4,4),NR(256),FA,FB,ND1(256),IV,NW
      COMMON /DIRAC/ IREC,JPOINT,NDM,LSAV
      NIN=5
      NOUT=18
      NW=30
      NDM=4
      LSAV=19
      IRAW=1
      NDX=5
      NVBL=1
      REWIND 8
      CALL PLOTS(IDUM,JDUM,8)
   10 CALL BEGIN(&20,&30)
   20 CALL INPUT(&30,&40)
   30 CALL EDIT
   40 IREC=1
      CALL YATES
   70 CALL BACK
   50 IREC=1
      CALL PAGIT(&30,&60,&70)
   60 CALL PLOTTR(&30,&50)
      END
```

YATES

MAIN PROGRAM

```
      SUBROUTINE BEGIN(*,*)
      EXTERNAL EOBINT
      COMMON KEY,IOVLY,ITYPE,JTYPE
      COMMON /DIRAC/ IREC,JPOINT,NDM,LSAV
      CALL GRINIT('B')
      CALL GCEOB(EOBINT)
      CALL MASKIT(1)
    1 CALL GRDPLY('     THIS UNIT WILL PERFORM AN ANALYSIS OF VARIANCE F
     1OR ANY',59,&1000)
      CALL GRDPLY('COMPLETE FACTORIAL DESIGN OF THE FORM',37,&1000)
      CALL GRDPLY(' ', 1,&1000)
      CALL GRDPLY('                                   R S T',35,&1000)
      CALL GRDPLY('                                  2 3 4        ',41,&1000)
      CALL GRDPLY(' ',1,&1000)
      CALL GRDPLY('                              WITH R+S+T  LESS THAN OR EQUA
     1L TO 4',59,&1000)
      CALL SPACE(1)
      CALL GRDPLY('     FOR UP TO 6 REPLICATIONS.',30,&1000)
      CALL SPACE(1)
      CALL GRDPLY('     IF DESIRED, THE UNIT WILL ALSO DETERMINE THE EQU
     1ATION OF',61,&1000)
      CALL GRDPLY('THE RESPONSE SURFACE AND, FURTHER, WILL ALLOW THE USE
     1R TO',57,&1000)
      CALL GRDPLY('EXAMINE PLOTS OF THIS SURFACE.',30,&1000)
      CALL SPACE(1)
      CALL GRDPLY('     AT ANY POINT DURING THE SESSION, THE USER MAY RE
     1TURN TO THE',64,&1000)
      CALL GRDPLY('GMS MONITOR BY DEPRESSING PROGRAM FUNCTION KEY 31.',5
     10,&1000)
      CALL SPACE(1)
      CALL GRDPLY('     PLEASE DEPRESS EITHER',26,&1000)
      CALL GRDPLY('          KEY 1 IF YOU WISH TO ENTER DATA, OR',45,&10
     100)
      CALL GRDPLY('          KEY 2 IF YOUR DATA ARE STORED ON DISK.',48,
     1&1000)
   24 CALL TWAIT(1,0,0,0)
      CALL GERAS(100)
      CALL GRRLSE
      IF(KEY.EQ.2) RETURN 2
      RETURN 1
 1000 CALL GERAS(100)
      GO TO 1
      END
```

```
      SUBROUTINE INPUT(*,*)
      EXTERNAL EOBINT
      DOUBLE PRECISION DUMMY
      DOUBLE PRECISION DSSQ,DER
      REAL NAME
      DIMENSION TEXT(20),JTYPE(4),NOP(6),ISET(6),REP(6)
      DIMENSION TEMTTL(20),TEMNAM(2)
      DIMENSION FAME(8)
      DIMENSION INHEAD(4),ABCD(4),IX(6),LTXT(20)
      COMMON KEY,IOVLY,ITYPE,JTYPE
      COMMON /BAKDAT/ A(256),VSE    ,SS(256),NER,SSE,SSER,NERR
      COMMON /YATDAT/ DSSQ,DER,
     3LVL(4),NFG,IRAW,DAT(256),NEND(4,4),TTL(20),
     1VNAM(2),NAME(8),FMT(80),ICODE,IL(6),NIN,NOUT,NDX,NF,
     2NVBL,NREP,CONCOM(4,4),NR(256),FA,FB,ND1(256),IV,NW
      COMMON /DIRAC/ IREC,JPOINT,NDM,LSAV
      DATA INHEAD/'A = ','B = ','C = ','D = '/
      DATA ABCD/'A','B','C','D'/
      DATA FAME/'FACA',' ','FACB',' ','FACC',' ','FACD',' '/
      DATA NOP/6*1/
      DATA ISET/6*1/
      DATA TEMTTL,TEMNAM/22*' '/
      DATA LBLNK/' '/
      DO 7 I=1,8
    7 NAME(I)=FAME(I)
      DO 9 I=1,4
      LVL(I)=1
      DO 9 J=1,4
    9 CONCOM(I,J)=J
      DO 11 IJ=1,256
      DAT(IJ)=0.
      NR(IJ)=0
   11 CONTINUE
      DSSQ=0.D0
      DER=0.D0
      ICOD=0
      VNAM(1)=TEMNAM(1)
      VNAM(2)=TEMNAM(2)
      DO 13 K=1,20
      TTL(K)=TEMTTL(K)
   13 CONTINUE
      NC=72
      NS=68
      CALL GRINIT('B')
      CALL GCEOB(EOBINT)
      CALL MASKIT(3)
   27 IERR=1
      CALL GRDPLY('PLEASE ENTER THE NUMBER OF FACTORS.',35,&1000)
      CALL TWAIT(0,0,1,0)
      CALL GRRPLX(TEXT,NC)
      CALL GRDPLY(TEXT,NC,&1000)
      INDEX=0
      CALL INX(TEXT,INDEX,NC,DUMMY,&900)
      NF=DUMMY+.001
```

```
      IF(NF.GT.4) GO TO 905
      CALL GRDPLY('PLEASE ENTER THE NUMBER OF REPLICATIONS.',40.&1000)
   29 IERR=2
      CALL TWAIT(0.0.1.0)
      CALL GRRPLX(TEXT.NC)
      CALL GRDPLY(TEXT.NC.&1000)
      INDEX=0
      CALL INX(TEXT.INDEX.NC.DUMMY.&900)
      NREP=DUMMY+.001
      IF(NREP.NE.1) GO TO 129
      CALL GRDPLY('ENTER AN ESTIMATE OF THE MSE.'.29.&1000)
      CALL TWAIT(0.0.1.0)
      CALL GRRPLX(TEXT.NC)
      INDEX=0
      CALL INX(TEXT.INDEX.NC.DUMMY.&900)
      VSE=DUMMY
      CALL GRDPLY('UPON HOW MANY DEGREES OF FREEDOM IS THE ESTIMATE BASE
     1D?'.55.&1000)
      CALL TWAIT(0.0.1.0)
      CALL GRRPLX(TEXT.NC)
      INDEX=0
      CALL INX(TEXT.INDEX.NC.DUMMY.&900)
      NER=DUMMY+.1
  129 IF(NREP.GT.6) GO TO 910
      CALL GRDPLY('DEPRESS KEY 1 TO CONTINUE.'.26.&900)
      CALL TWAIT(1.0.0.0)
      CALL MASKIT(3)
      CALL GERAS(100)
      CALL GRDPLY('DEPRESS KEY 1 TO USE DEFAULT NAMES FACA. FACB. ETC.'.
     151.&1000)
      CALL GRDPLY('OR ENTER THE NAME OF THE FIRST FACTOR (LIMIT 8 LETTER
     1S).'.56.&1000)
      CALL TWAIT(1.0.1.0)
      IF(ITYPE.EQ.1) GO TO 100
      CALL GRRPLX(TEXT.NC)
      CALL GRDPLY(TEXT.NC.&1000)
      NAME(1)=TEXT(1)
      NAME(2)=TEXT(2)
      NUP=2*NF-3
      IF(NF.EQ.2) GO TO 60
      DO 50 I=3.NUP.2
      CALL GRDPLY('PLEASE ENTER THE NAME OF THE NEXT FACTOR.'.41.&1000)
      CALL TWAIT(0.0.1.0)
      CALL GRRPLX(TEXT.NC)
      CALL GRDPLY(TEXT.NC.&1000)
      NAME(I)=TEXT(1)
      NAME(I+1)=TEXT(2)
   50 CONTINUE
   60 CALL GRDPLY('PLEASE ENTER THE NAME OF THE LAST FACTOR.'.41.&1000)
      CALL TWAIT(0.0.1.0)
      CALL GRRPLX(TEXT.NC)
      CALL GRDPLY(TEXT.NC.&1000)
      NAME(NUP+2)=TEXT(1)
      NAME(NUP+3)=TEXT(2)
```

```
100 CONTINUE
 31 IERR=3
    CALL OERAS(100)
    CALL GRDPLY('THERE WILL BE AN OPPORTUNITY AT THE END OF THIS FRAME
   1'.53.&1000)
    CALL GRDPLY('TO CORRECT ANY INPUT ERRORS.'.28.&1000)
    NFG=1
    DO 110 I=1.NF
    J=2*(I-1)+1
    WRITE(NDM.500) NAME(J).NAME(J+1)
    CALL FETCH(TEXT.MC)
    CALL GRDPLY(TEXT.MC.&1000)
108 CALL TWAIT(0.0.1.0)
    CALL GRRPLX(TEXT.NC)
    INDEX=0
    CALL INX(TEXT.INDEX.NC.DUMMY.&900)
    LVL(I)=DUMMY+.0001
    IF(LVL(I).GT.1.AND.LVL(I).LT.5) GO TO 109
    CALL GRDPLY('ILLEGAL VALUE.  PLEASE RE-ENTER'.31.&1000)
    GO TO 108
109 NFG=NFG*LVL(I)
    WRITE(NDM.510) I.LVL(I)
    CALL FETCH(TEXT.MC)
    CALL GRDPLY(TEXT.MC.&1000)
    CALL SPACE(1)
110 CONTINUE
    CALL MASKIT(1)
    CALL GRDPLY('DEPRESS KEY 1 TO CONTINUE OR KEY 2 TO RE-ENTER LEVELS
   1.'.52.&1000)
    CALL TWAIT(1.0.0.0)
    IF(KEY.EQ.2) GO TO 100
    CALL OERAS(100)
    CALL GRDPLY('      IF YOU WISH THE LEVELS OF THE FACTORS TO BE'.48.
   1&1000)
    CALL GRDPLY('1 FOR LEVEL 1. 2 FOR LEVEL 2. ETC.  DEPRESS KEY 1'.
   149.&1000)
    CALL SPACE(1)
    CALL GRDPLY('      IF YOU WISH TO ENTER VALUES FOR THE LEVELS. DEPR
   1ESS KEY 2'.62.&1000)
    CALL TWAIT(1.0.0.0)
    IF(KEY.EQ.1) GO TO 200
    DO 150 I=1.NF
125 CALL SPACE(1)
    J=2*(I-1)+1
    WRITE(NDM.520) NAME(J).NAME(J+1)
520 FORMAT('ENTER. SEPARATED BY COMMAS. THE LEVELS FOR '.2A4)
    CALL FETCH(TEXT.MC)
    CALL GRDPLY(TEXT.MC.&2000)
155 CALL TWAIT(0.0.1.0)
    CALL GRRPLX(TEXT.NC)
    CALL SPACE(1)
    CALL GRDPLY(TEXT.NC.&2000)
    INDEX=0
    NL=LVL(I)
```

```
      DO 135 K=1,NL
      IF(INDEX.LT.0) GO TO 855
      CALL INX(TEXT,INDEX,NC,DUMMY,&850)
      CONCOM(I,K)=DUMMY
  135 CONTINUE
      GO TO 150
  855 CALL GRDPLY('THERE ARE LESS NUMBERS THAN EXPECTED.  PLEASE RE-ENTE
     1R LINE',59,&1000)
      GO TO 155
  850 CALL  GRDPLY('AN ERRONEOUS NUMBER WAS ENTERED.  DEPRESS KEY 2 TO T
     1RY AGAIN.',61,&1000)
      CALL TWAIT(1,0,0,0)
      GO TO 125
  150 CONTINUE
  200 CALL SPACE(1)
      CALL GRDPLY('DEPRESS KEY 1 TO CONTINUE',25,&1000)
      CALL TWAIT(1,0,0,0)
      CALL GERAS(100)
      CALL MASKIT(10)
      CALL GRDPLY('      ENTER, SEPARATED BY COMMAS, ALL REPLICATIONS FOR
     1',53,&2000)
      CALL GRDPLY('THE FACTOR LEVELS INDICATED',27,&2000)
      CALL SPACE(1)
      WRITE(NOM,540) (INHEAD(KOUT),NAME(2*KOUT-1),NAME(2*KOUT),KOUT=1,
     1NF)
      CALL FETCH(TEXT,MC)
      CALL GRDPLY(TEXT,MC,&2000)
      CALL SPACE(1)
      CALL GRDPLY('      DEPRESS KEY 30 TO EDIT A LINE OF DATA.',43,
     1&1000)
      CALL SPACE(1)
      WRITE(NOM,545) (ABCD(KOUT),KOUT=1,NF)
      CALL FETCH(TEXT,MC)
      CALL GRDPLY(TEXT,MC,&2000)
      CALL SPACE(1)
  540 FORMAT(4(3A4,4X))
  545 FORMAT('CELL',4(1X,A1))
      DO 107 KBL=1,4
      LTXT(KBL)=LBLNK
  107 CONTINUE
      DO 401 I=1,NFG
  402 IF(I.GT.36) CALL GERAS(1)
      CALL ANAL(I,IX(1),IX(2),IX(3),IX(4))
      WRITE(NOM,560) I,(IX(KOUT),KOUT=1,NF)
      CALL FETCH(LTXT,MC)
      CALL GRDPLY(LTXT,MC,&2000)
  560 FORMAT(I2,2X,4I2)
  411 CALL TWAIT(1,0,1,0)
      IF(ITYPE.EQ.1) GO TO 700
      CALL GRRPLX(TEXT,NC)
      INDEX=0
      DO 406 J=1,NREP
      IF(INDEX.LT.0) GO TO 408
      CALL INX(TEXT,INDEX,NC,DUMMY,&853)
```

```
      DAT(I)=DAT(I)+DUMMY
      DSSQ=DSSQ+DUMMY*DUMMY
      REP(J)=DUMMY
      GO TO 406
  853 CALL GRDPLY('AN ERRONEOUS NUMBER WAS ENTERED.  TRY AGAIN.',44,
     1&1000)
      IF(J.EQ.1) GO TO 411
      DAT(I)=0.
      JMO=J-1
      DO 854 IJ=1,JMO
      DSSQ=DSSQ-REP(IJ)*REP(IJ)
  854 CONTINUE
      GO TO 411
  406 CONTINUE
      WRITE(LSAV'I) REP
      CALL GBKSP(1)
      WRITE(NDM,570) (LTXT(KOUT),KOUT=1,3),(TEXT(JOUT),JOUT=1,15)
  570 FORMAT(3A4,1X,15A4)
      CALL FETCH(TEXT,MC)
      CALL GRDPLY(TEXT,MC,&2000)
      GO TO 401
  408 WRITE(NDM,575) NREP
  575 FORMAT('YOU DID NOT ENTER ',I1,' REPLICATIONS.  RE-ENTER LINE')
      CALL FETCH(TEXT,MC)
      CALL GRDPLY(TEXT,MC,&2000)
      GO TO 411
  700 IF(I.GT.32) CALL GERAS(4)
      CALL GRDPLY('ENTER THE NUMBER OF THE CELL THAT YOU WISH TO EDIT.',
     151,&2000)
  437 CALL TWAIT(0,0,1,0)
      CALL GRRPLX(TEXT,NC)
      INDEX=0
      CALL INX(TEXT,INDEX,NC,DUMMY,&924)
      GO TO 925
  924 CALL GRDPLY('RE-ENTER CELL NUMBER.',21,&1000)
      GO TO 437
  925 IEDIT=DUMMY+.05
      READ(LSAV'IEDIT) REP
      DO 418 JJ=1,NREP
      DAT(IEDIT)=DAT(IEDIT)-REP(JJ)
      DSSQ=DSSQ-REP(JJ)*REP(JJ)
  418 CONTINUE
      CALL ANAL(IEDIT,IX(1),IX(2),IX(3),IX(4))
      CALL GRDPLY('ENTER THE NEW DATA',18,&2000)
      WRITE(NDM,560) IEDIT,(IX(KOUT),KOUT=1,NF)
      CALL FETCH(LTXT,MC)
      CALL GRDPLY(LTXT,MC,&2000)
  431 CALL TWAIT(0,0,1,0)
      CALL GRRPLX(TEXT,NC)
      CALL GBKSP(2)
      WRITE(NDM,570)(LTXT(KOUT),KOUT=1,3),(TEXT(JOUT),JOUT=1,15)
      CALL FETCH(LTXT,MC)
      CALL GRDPLY(LTXT,MC,&2000)
      INDEX=0
```

```
      DO 426 J=1,NREP
      IF(INDEX.LT.0) GO TO 428
      CALL INX(TEXT,INDEX,NC,DUMMY,&859)
      REP(J)=DUMMY
      DAT(IEDIT)=DAT(IEDIT)+DUMMY
      DSSQ=DSSQ+DUMMY*DUMMY
      GO TO 426
  859 CALL GRDPLY('AN ERRONEOUS NUMBER WAS ENTERED.  TRY AGAIN.',44,
     1&1000)
      IF(J.EQ.1) GO TO 431
      DAT(IEDIT)=0.
      JM0=J-1
      DO 856 IJ=1,JM0
      DSSQ=DSSQ-REP(IJ)*REP(IJ)
  856 CONTINUE
      GO TO 431
  426 CONTINUE
      WRITE(LSAV'IEDIT) REP
      GO TO 402
  428 WRITE(NDM,575) NREP
      CALL FETCH(TEXT,MC)
      CALL GRDPLY(TEXT,MC,&2000)
      GO TO 431
  401 CONTINUE
      CALL GERAS(3)
      CALL MASKIT(1)
      CALL GRDPLY('DATA ENTRY IS COMPLETE.',23,&1000)
      CALL GRDPLY('DEPRESS KEY 1 TO PERFORM ANALYSIS',33,&1000)
      CALL GRDPLY('DEPRESS KEY 2 TO REVIEW OR EDIT DATA.',37,&1000)
      CALL TWAIT(1,0,0,0)
      WRITE(LSAV'257) NF,NREP,LVL
      WRITE(LSAV'258) DER,DSSQ,NFG
      WRITE(LSAV'260) NAME,CONCOM,VSE
      IF(KEY.EQ.1) GO TO 550
      CALL GRRLSE
      RETURN 1
  550 CALL GERAS(100)
      CALL MASKIT(1)
      CALL SPACE(1)
      CALL GRDPLY('DEPRESS KEY 1 TO DELETE EACH SINGLE DEGREE OF FREEDOM
     1 F LESS THAN 1.0',69,&1000)
      CALL GRDPLY('DEPRESS KEY 2 TO RETAIN ALL F VALUES',36,&1000)
      CALL TWAIT(1,0,0,0)
      FA=1.0
      IF(KEY.EQ.2) FA=0.
      CALL GERAS(100)
      CALL GRRLSE
      RETURN 2
 2000 CONTINUE
 1000 WRITE(6,530)
  530 FORMAT('1SCREEN FULL BUG')
      CALL GRRLSE
      RETURN
  900 CALL GERAS(100)
```

```
      CALL GRDPLY('AN ERRONEOUS NUMBER WAS ENTERED.  PLEASE TRY AGAIN.',
     151,&1000)
      GO TO (27,29,31),IERR
  905 CALL GERAS(100)
      CALL GRDPLY('THE NUMBER OF FACTORS MUST BE LESS THAN 5.',42,&1000)
      GO TO 27
  910 CALL GERAS(100)
      CALL GRDPLY('THE NUMBER OF REPLICATIONS MUST BE LESS THAN 7.',47,&
     11000)
      GO TO 29
  500 FORMAT('PLEASE ENTER THE NUMBER OF LEVELS FOR ',2A4)
  510 FORMAT('FACTOR',I2,' HAS',I2,' LEVELS.')
      END
```

```
      SUBROUTINE EDIT
      EXTERNAL EOBINT
      DOUBLE PRECISION DUMMY,DSSQ,DER
      REAL NAME
      DIMENSION TEXT(20),JTYPE(4),NOP(6),ISET(6),REP(6)
      COMMON KEY,IOVLY,ITYPE,JTYPE
      COMMON /BAKDAT/ A(256),VSE    ,SS(256),NER,SSE,SSER,NERR
      COMMON /YATDAT/ DSSQ,DER,
     3LVL(4),NFC,IRAW,DAT(256),NEND(4,4),TTL(20),
     1VNAM(2),NAME(8),FMT(80),ICODE,IL(6),NIN,NOUT,NDX,NF,
     2NVBL,NREP,CONCOM(4,4),NR(256),FA,FB,ND1(256),IV,NW
      COMMON /DIRAC/ IREC,JPOINT,NDM,LSAV
      DIMENSION INHEAD(4),ABCD(4),IX(6),LTXT(4)
      DATA INHEAD /'A =','B =','C =','D ='/
      DATA ABCD /'A','B','C','D'/
      DATA LBLNK/' '/
      NC=72
      READ(LSAV'257) NF,NREP,LVL
      READ(LSAV'258) DER,DSSQ,NFC
      READ(LSAV'260) NAME,CONCOM,VSE
      CALL GRINIT('B')
      CALL GCEOB(EOBINT)
   50 CALL GRDPLY('      DEPRESS',12,&1000)
      CALL GRDPLY('           KEY 1 TO CONTINUE WITH THE ANALYSIS',45,
     1&1000)
      CALL GRDPLY('           KEY 2 TO VIEW THE RAW DATA',36,&1000)
      CALL GRDPLY('           KEY 3 TO REPLACE DATA',31,&1000)
      CALL MASKIT(4)
      CALL TWAIT(1,0,0,0)
      CALL GERAS(100)
      IF(KEY.EQ.1) GO TO 900
      IF(KEY.EQ.2) GO TO 500
      CALL SPACE(1)
   55 CALL GRDPLY('       ENTER THE NUMBER OF THE CELL THAT YOU WISH TO R
     1EPLACE.',60,&1000)
   90 CALL TWAIT(0,0,1,0)
      CALL GRRPLX(TEXT,NC)
      INDEX=0
      CALL INX(TEXT,INDEX,NC,DUMMY,&1110)
      GO TO 92
 1110 CALL GRDPLY('AN ERRONEOUS NUMBER WAS ENTERED.  PLEASE TRY AGAIN.',
     151,&1000)
      GO TO 90
   92 NCELL=DUMMY+.5
      IF(NCELL.GT.0.AND.NCELL.LE.NFC) GO TO 100
      CALL SPACE(1)
      CALL GRDPLY('THE CELL NUMBER ENTERED IS OUT OF THE PROPER RANGE.
     1TRY AGAIN',62,&1000)
      GO TO 90
  100 CALL GRDPLY('      ENTER, SEPARATED BY COMMAS, ALL REPLICATIONS FOR
     1',53,&1000)
      CALL GRDPLY('THE FACTOR LEVELS INDICATED',27,&1000)
      CALL SPACE(1)
      WRITE(NDM,540)(INHEAD(KOUT),NAME(2*KOUT-1),NAME(2*KOUT),KOUT=1,NF)
```

```
540 FORMAT(4(3A4.4X))
    CALL FETCH(TEXT.MC)
    CALL GRDPLY(TEXT.MC.&1000)
    CALL SPACE(1)
    WRITE(NDM.545)(ABCD(KOUT).KOUT=1.NF)
    CALL FETCH(TEXT.MC)
    CALL ORDPLY(TEXT.MC.&1000)
    CALL SPACE(1)
    CALL ANAL(NCELL.IX(1).IX(2).IX(3).IX(4))
    WRITE(NDM.560) NCELL.(IX(KOUT).KOUT=1.NF)
    CALL FETCH(LTXT.MC)
    CALL GRDPLY(LTXT.MC.&1000)
560 FORMAT(I2.2X.4I2)
105 CALL TWAIT(0.0.1.0)
    CALL GRRPLX(TEXT.NC)
    INDEX=0
    DO 120 J=1.NREP
    IF(INDEX.LT.0) GO TO 130
    CALL INX(TEXT.INDEX.NC.DUMMY.&1100)
    REP(J)=DUMMY
    GO TO 120
1100 CALL GRDPLY('AN ERRONEOUS NUMBER WAS ENTERED.  PLEASE TRY AGAIN.'.
   151.&1000)
    GO TO 105
120 CONTINUE
    WRITE(LSAV'NCELL) REP
    GO TO 140
130 CALL SPACE(1)
    WRITE(NDM.575) NREP
575 FORMAT('YOU DID NOT ENTER '.I1.' REPLICATIONS.  RE-ENTER LINE.')
    CALL FETCH(TEXT.MC)
    CALL ORDPLY(TEXT.MC)
    GO TO 105
140 CALL GBKSP(1)
    WRITE(NDM.570) (LTXT(KOUT).KOUT=1.3).(TEXT(JOUT).JOUT=1.15)
570 FORMAT(3A4.1X.15A4)
    CALL FETCH(TEXT.MC)
    CALL GRDPLY(TEXT.MC.&1000)
    CALL SPACE(1)
    GO TO 50
500 LOW=1
501 CALL GRDPLY('     DEPRESS'.12.&1000)
    CALL GRDPLY('            KEY 1 TO CONTINUE WITH THE ANALYSIS'.45.
   1&1000)
    CALL GRDPLY('            KEY 3 TO REPLACE DATA'.31.&1000)
    CALL GRDPLY('DEPRESS KEY 28 TO PAGE BACKWARD.'.32.&1000)
    CALL GRDPLY('DEPRESS KEY 29 TO PAGE FORWARD.'.31.&1000)
    CALL MASKIT(11)
    CALL SPACE(1)
    WRITE(NDM.540) (INHEAD(KOUT).NAME(2*KOUT-1).NAME(2*KOUT).KOUT=1.
   1NF)
    CALL FETCH(TEXT.MC)
    CALL ORDPLY(TEXT.MC)
    CALL SPACE(1)
```

```
      WRITE(NDM.545) (ABCD(KOUT).KOUT=1.NF)
      CALL FETCH(TEXT.MC)
      CALL GRDPLY(TEXT.MC.&1000)
      CALL SPACE(1)
  545 FORMAT('CELL',4(1X.A1))
      DO 107 KBL=1.4
      LTXT(KBL)=LBLNK
  107 CONTINUE
      JHI=LOW+35
      JHI=MINO(JHI.NFG)
      DO 600 J=LOW.JHI
      CALL ANAL(J.IX(1).IX(2).IX(3).IX(4))
      WRITE(NDM.560) J.(IX(KOUT).KOUT=1.NF)
      CALL FETCH(LTXT.MC)
      READ(LSAV'J) REP
      WRITE(NDM.572)(LTXT(KOUT).KOUT=1.3).(REP(JOUT).JOUT=1.NREP)
  572 FORMAT(3A4.6G10.4)
  580 FORMAT(18A4)
      CALL FETCH(TEXT.MC)
      CALL GRDPLY(TEXT.MC.&1000)
  600 CONTINUE
      CALL TWAIT(1.0.0.0)
      CALL GERAS(100)
      IF(KEY.EQ.1) GO TO 900
      IF(KEY.EQ.3) GO TO 55
      IF(KEY.EQ.28) LOW=MAXO(1.LOW-36)
      IF(KEY.EQ.29) LOW=LOW+36
      IF(LOW.GT.NFG) LOW=LOW-36
      GO TO 501
  900 CALL GERAS(100)
      CALL MASKIT(1)
      CALL SPACE(1)
      CALL GRDPLY('DEPRESS KEY 1 TO DELETE EACH SINGLE DEGREE OF FREEDOM
     1 F LESS THAN 1.0',69.&1000)
      CALL GRDPLY('DEPRESS KEY 2 TO RETAIN ALL F VALUES',36.&1000)
      CALL TWAIT(1.0.0.0)
      FA=1.0
      IF(KEY.EQ.2) FA=0.
      CALL GERAS(100)
      CALL GRRLSE
      DSSQ=0.
      DO 910 K=1.NFG
      DAT(K)=0.
      READ(LSAV'K) REP
      DO 920 J=1.NREP
      DAT(K)=DAT(K)+REP(J)
      DSSQ=DSSQ+REP(J)*REP(J)
  920 CONTINUE
  910 CONTINUE
      RETURN
 1000 WRITE(6.530)
  530 FORMAT('1SCREEN FULL BUG')
      CALL GRRLSE
      RETURN
      END
```

```
      SUBROUTINE  YATES
      DOUBLE PRECISION DSSQ.DER.DSET.DA.DB.DC
      COMMON /DIRAC/ IREC.JPOINT.NDM.LSAV
      COMMON /YATDAT/ DSSQ(1).DER(1).
     3        LVL(4).NFG.IRAW.DAT(256.1).NEND(4.4).TTL(20).VNAM(2.1).FNA
     1M(2.4).FMT(80).ICOD(1).IL(6).                   NIN.NOUT.NDX.NFAC.NVBL
     2.NREP.RAW(4.4.1).NR(256).FA.FB.ND1(256).IV.NW
      COMMON /BAKDAT/ A(256).VSE(1).SS(256).NER.SSE.SSER.NERR
      DIMENSION SQA(64).SETUP(38).ISET(38)
      EQUIVALENCE (SETUP.ISET)
      KPG = 0
      IF(NREP.EQ.1) GO TO 10
      DO 2 J=1.NVBL
      DA = 0.D0
      DSET = 0.D0
      DO 1 I=1.NFG
      DB = DAT(I.J)
      DA = DA + DB**2
1     DSET = DSET + DB
      DB = NREP
      DER(J) = DSSQ(J) - DA/DB
      DA = (NREP-1)*NFG
      VSE(J) = DER(J)/DA
2     CONTINUE
81    FORMAT (I3.8F8.3)
10    DO 201 NC=1.NFG
      CALL ANAL(NC.IL(1).IL(2).IL(3).IL(4))
      ND = 1
      DO 202 JJ=1.NFAC
      IF(IL(JJ).EQ.1) ND = ND*LVL(JJ)
202   CONTINUE
      ND = ND*NREP
      ND1(NC) = ND
      ND = 1
      DO 203 JK = 1.NFAC
      IF(IL(JK).EQ.4) ND = ND*20
      IF(LVL(JK)-3) 204.204.206
204   IF(IL(JK).EQ.2) ND = ND*2
      IF(IL(JK).EQ.3) ND = ND*6
      GO TO 203
206   IF(IL(JK).EQ.2) ND = ND*20
      IF(IL(JK).EQ.3) ND = ND*4
203   CONTINUE
      ND1(NC) = ND1(NC)*ND
201   CONTINUE
      DO 200 IW=1.NVBL
      KPG = 0
      IV = IW
      DO 105 IU=1.NFAC
      ISX = NFG/LVL(IU)
      LV = LVL(IU)
      KDK = 1
      DO 15 JA=1.ISX
15    A(JA) = 0.
```

```
      DO 16 MV=1.ISX
      DO 16 JB=1.LV
      A(MV) = A(MV) + DAT(KDK.IV)
 16   KDK = KDK+1
      KDK = 1
      GO TO (12.12.13.14).LV
 12   JBA = ISX + 1
      JBB = 2*ISX
      DO 22 JBX = JBA.JBB
      A(JBX) = DAT(KDK+1.IV) - DAT(KDK.IV)
 22   KDK = KDK + 2
      GO TO 100
 13   JBA = ISX+1
      JBB = 2*ISX
      DO 32 JBY = JBA.JBB
      A(JBY) = DAT(KDK+2,IV) - DAT(KDK.IV)
 32   KDK = KDK+3
      KDK = 1
      JBA = 2*ISX + 1
      JBB = 3*ISX
      DO 33 JBZ = JBA.JBB
      A(JBZ) = DAT(KDK+2,IV)+DAT(KDK.IV)-2.*DAT(KDK+1.IV)
 33   KDK = KDK + 3
      GO TO 100
 14   JBA = ISX + 1
      JBB = 2*ISX
      KDK = 1
      DO 42 JBP = JBA.JBB
      A(JBP) = 3.*(DAT(KDK+3.IV)-DAT(KDK.IV))+DAT(KDK+2.IV)-DAT(KDK+1.IV
     1)
 42   KDK = KDK+4
      KDK = 1
      JBA = 2*ISX+1
      JBB = 3*ISX
      DO 43 JBQ = JBA.JBB
      A(JBQ) =DAT(KDK.IV)-DAT(KDK+1,IV)-DAT(KDK+2.IV)+DAT(KDK+3.IV)
 43   KDK = KDK+4
      JBA = 3*ISX+1
      JBB = 4*ISX
      KDK = 1
      DO 44 JBS=JBA.JBB
      A(JBS)=3.*(DAT(KDK+1.IV)-DAT(KDK+2,IV))+DAT(KDK+3,IV)-DAT(KDK.IV)
 44   KDK = KDK+4
 100  DO 107 JBT=1.NFG
 107  DAT(JBT.IV) = A(JBT)
 105  CONTINUE
      DO 110 KA=1.NFG
      AD = ND1(KA)
      A(KA) = DAT(KA.IV)/AD
      NR(KA) = 0
 110  SS(KA) = DAT(KA,IV)**2/AD
C  ANOVA TABLE
      KBE = 2**NFAC - 1
      DO 114 KB=1.KBE
```

```
 114  SQA(KB) = 0.
      SQAL = 0.
      DO 115 KC=1,NFG
      CALL ANAL(KC,IL(1),IL(2),IL(3),IL(4))
      KDA = 0
      DO 116 KD=1,NFAC
      KDA = 2*KDA
      KDX = NFAC-KD+1
      IF(IL(KDX).GT.1) KDA=KDA+1
 116  CONTINUE
      IF(KDA.EQ.0) GO TO 115
      NR(KDA) = NR(KDA) + 1
      SQAL = SQAL + SS(KC)
      SQA(KDA) = SQA(KDA) + SS(KC)
 115  CONTINUE
      IRET = 1
 125  CONTINUE
      GO TO (119,121),IRET
 119  DO 120 KG=1,NFAC
      NR(KG+65) = LVL(KG)
 120  LVL(KG) = 2
      KH = 1
127   CALL ANAL(KH+1,IL(1),IL(2),IL(3),IL(4))
      IX = 0
      DO 122 KI=1,NFAC
      IF(IL(KI)-1) 122,122,124
 124  IX=IX+1
      ISET(IX) = KI
 122  CONTINUE
      IX2 = 2*IX
      DO 118  KL=1,IX
      KA=ISET(KL)
      SETUP(2*KL+6) = FNAM(1,KA)
 118  SETUP(2*KL+7) = FNAM(2,KA)
      ADF=NR(KH)
      VSH=SQA(KH)/ADF
      FF=0.
      IF(VSE(IV).LE.0.)  GO TO 128
      FF=VSH/VSE(IV)
 128  KMB=IX2 + 7
      WRITE(NOUT'IREC,339)(SETUP(KM),KM=8,KMB)
 339  FORMAT(1H0,2A4,1H*,5(2A4,1H*))
      WRITE(NOUT'IREC,341) NR(KH),SQA(KH),VSH,FF
 341  FORMAT(1H ,I3,1X,2G12.6,G12.4)
      KPG = KPG+3
      IRET = 2
      IF(KPG.GT.45) KPG=0
      IF(KPG.EQ.0) GO TO 125
 121  KH=KH+1
      IF(KH.LE.KBE) GO TO 127
      DO 828 KN=1,NFAC
 828  LVL(KN) = NR(KN+65)
      IF(NREP.EQ.1)  GO TO 1166
      NER = (NREP-1)*NFG
```

```
1166 NALL = NFG-1
     VSH = SQAL/FLOAT(NALL)
     FF = 0.
     IF(VSE(IV).EQ.0.) GO TO 151
     FF = VSH/VSE(IV)
2013 FORMAT(55X)
     WRITE(NOUT'IREC.2013)
     WRITE(NOUT'IREC.2013)
 151 WRITE(NOUT'IREC.350)
350  FORMAT(1H .19X.11HALL EFFECTS.22X)
     WRITE(NOUT'IREC.341) NALL.SQAL.VSH.FF
     WRITE(NOUT'IREC.2013)
     WRITE(NOUT'IREC.2013)
     WRITE(NOUT'IREC.351)
351  FORMAT(1H .22X.5HERROR.25X)
     SSE = DER(IV)
     IF(NREP.EQ.1)  SSE = VSE(IV)*FLOAT(NER)
     WRITE(NOUT'IREC.3411) NER.SSE.VSE(IV)
3411 FORMAT(1H .I3.1X.2G12.6.12X)
     WRITE(NOUT'IREC.1000)
1000 FORMAT(' LAST')
     SSER = SSE
     NERR = NER
     DO 162 KP=1.NFG
     IF(SS(KP)/VSE(IV).GE.FA)  GO TO 162
     A(KP) = 0.
     SSE = SSE + SS(KP)
     NER = NER + 1
162  CONTINUE
     IREC=70
     REWIND NW
     WRITE(NW) A.SS.SSE.NER,VSE
200  CONTINUE
99   REWIND NW
     RETURN
1313 FORMAT(' LAST')
     END
```

```
      SUBROUTINE BACK
      DOUBLE PRECISION DSSQ.DER.DSET.DA.DB
      COMMON /DIRAC/ IREC
      COMMON /YATDAT/ DSSQ(1).DER(1).
     3         LVL(4).NFG.IRAW.DAT(256.1).NEND(4.4).TTL(20).VNAM(2.1).FNA
     1M(2.4).FMT(80).ICOD(1).IL(6).                 NIN.NOUT.NDX.NFAC.NVBL
     2.NREP.RAW(4.4.1).NR(256).FA.FB.ND1(256).IV.NW.COEF(4.6).ASAV(512)
      COMMON /BAKDAT/ A(256).VSE    .SS(256).NER.SSE.SSER.NERR
      DIMENSION SETUP(45).ISET(45).CC(6).CEX(3).SCAL(20).
     1FMG(40).AR(6).CMP(3).AN(5).BN(5).CONV(6.4)
      DIMENSION BLANK(50)
      EQUIVALENCE (SETUP.ISET)
      DATA FILL/4H 2X./
      DATA  CC /2H*A.2H*B.2H*C.2H*D.2H*E.2H*F/
      DATA  CEX /4H**2 .4H**3 .4H**4 /
      DATA  BL/4H    /
      DATA  FMG(1)/4H(1H0/
      DATA PLS/4H1H+./
      DATA  EA.EB/4H  G1.4H4.7./
      DATA BLANK/50*' '/
      DATA FARG/4H A3./
      DATA FARG1/4H50A1/
      DATA FARG2/4H    )/
      DATA  ARG/4H A2./
      DATA ARGP/4H A4./
      DATA AR/4HARAW.4HBRAW.4HCRAW.4HDRAW.4HERAW.4HFRAW/
      IREC=70
      KPG = 1
      DO 1 I=1.NFG
1     DAT(I.IV) = A(I)
      VSE = SSE/FLOAT(NER)
      IF(SSE.LT.1.E-7) SSE = 0.
      VSE = SSE/FLOAT(NER)
      IF(SSE.GT.0.) GO TO 369
      RMS = 0.
      RMSM = 0.
      GO TO 379
369   ANT = NFG*NREP
      ADD = 1. + 1./ANT
      RMS = 2.*SQRT(VSE*ADD)
      DO 341 I=1.NFG
      IF(A(I).EQ.0.) GO TO 341
      CALL ANAL(I.IL(1).IL(2).IL(3).IL(4))
      NF = 1
      DO 342 IC=1.NFAC
      IF(IL(IC).EQ.4) NF = NF*9
342   CONTINUE
      ANF = NF
      AD = ND1(I)
      ADD = ADD + ANF/AD
341   CONTINUE
      RMSM = 2.*SQRT(VSE*ADD)
  379 CONTINUE
      DO 105 IU=1.NFAC
```

```
      ISX = NFG/LVL(IU)
      LV = LVL(IU)
      KDK = 1
      DO 15 JA=1,ISX
15    A(JA)=0.
      JBA = ISX+1
      JBB = 2*ISX
      JEA = JBB+1
      JEB = 3*ISX
      JFA = JEB+1
      JFB = 4*ISX
      GO TO (12,12,13,14),LV
12    DO 16 JB=1,ISX
      A(JB) = DAT(KDK,IV)
16    KDK = KDK+2
      KDK = 1
      DO 22 JC=JBA,JBB
      A(JC) = DAT(KDK+1,IV)
22    KDK=KDK+2
      GO TO 100
13    DO 31 JD=1,ISX
      A(JD) = DAT(KDK,IV)-2.*DAT(KDK+2,IV)
31    KDK = KDK+3
      KDK=1
      DO 32 JE=JBA,JBB
      A(JE) = DAT(KDK+1,IV)
32    KDK = KDK+3
      KDK=1
      DO 33 JF=JEA,JEB
      A(JF) = 3.*DAT(KDK+2,IV)
33    KDK = KDK+3
      GO TO 100
14    DO 41 JG=1,ISX
      A(JG) = DAT(KDK,IV)-1.25*DAT(KDK+2,IV)
41    KDK = KDK+4
      KDK=1
      DO 42 JH=JBA,JBB
      A(JH)=(DAT(KDK+1,IV)-41.*DAT(KDK+3,IV)/12.)*3.
42    KDK=KDK+4
      KDK=1
      DO 43 JI=JEA,JEB
      A(JI)=2.25*DAT(KDK+2,IV)
43    KDK=KDK+4
      KDK = 1
      DO 44 JK=JFA,JFB
      A(JK)=45.*DAT(KDK+3,IV)/4.
44    KDK=KDK+4
100   DO 107 JM=1,NFG
107   DAT(JM,IV)=A(JM)
105   CONTINUE
      DO 509 JJPB=1,NFG
      ASAV(JJPB)=A(JJPB)
 509  CONTINUE
      KPG = 0
```

```
      IRET = 1
      WRITE(NOUT'IREC.82) (CC(JA).FNAM(1.JA).FNAM(2.JA).JA=1.NFAC)
82    FORMAT(1H0.20X.6HFACTOR.A3.4H  IS.2X.2A4/(21X.6HFACTOR.A3.4H  IS.
     12X.2A4))
      IF(NER.EQ.0)  GO TO 1081
      VSSE = SSE/FLOAT(NER)
1081  WRITE(NOUT'IREC.2013)
2013  FORMAT(72X)
      WRITE(NOUT'IREC,83)
83    FORMAT(1H0.10X.30HIN STANDARD FACTOR VALUES. Y =  )
      DO 203 JA=1,45
203   SETUP(JA) = BL
      DO 230 JAA=2.40
230   FMG(JAA) = BL
      KDL = 2
      KDK = 1
      LCT = 0
      GO TO (501.502).IRET
501   DO 201 KH=1.NFG
      CALL ANAL(KH.IL(1).IL(2).IL(3).IL(4))
      DO 202 KG=1.6
202   IL(KG) = IL(KG) -1
      IF(ABS(A(KH)).LE.1.E-5) GO TO 2201
      FMG(KDL) = FILL
      KDL = KDL + 1
      SETUP(KDK)=A(KH)
      IF(A(KH)) 221.201.222
222   FMG(KDL)=PLS
      KDL=KDL+1
221   FMG(KDL)=EA
      KDL = KDL + 1
      FMG(KDL) = EB
      KDL=KDL+1
      KDK=KDK+1
      LCT=LCT+15
      DO 210 MA=1.NFAC
      IF(IL(MA)-1) 210.212,213
212   SETUP(KDK) = CC(MA)
      FMG(KDL) = ARG
      KDL = KDL+1
      KDK = KDK + 1
      LCT = LCT + 2
      GO TO 210
213   SETUP(KDK) = CC(MA)
      FMG(KDL) = ARG
      KDL=KDL+1
      MB=IL(MA)-1
      KDK=KDK+1
      SETUP(KDK) = CEX (MB)
      FMG(KDL) = ARGP
      KDL = KDL+1
      LCT = LCT+6
      KDK=KDK+1
210   CONTINUE
```

```
2201 IF(LCT.LE.30 .AND. KH.LT.NFG)  GO TO 201
     IF(KDL.LE.4 .OR. KDK.LE.2) GO TO 201
     KDL=KDL-1
     FMG(KDL) = FARG
     FMG(KDL+1)=FARG1
     FMG(KDL+2)=FARG2
     JJPBUP=KDL+2
     WRITE(6.1323) (FMG(JJPB).JJPB=1.JJPBUP)
1323 FORMAT(10X.40A4)
     MPAT=68-LCT
     KDK=KDK-1
     WRITE(NOUT'IREC.FMG) (SETUP(MD).MD=1.KDK).
    1(BLANK(KPAT).KPAT=1.MPAT)
     KPG=KPG+1
     IF(KPG.GT.45) KPG=0
     KPG = 1
     KDK=1
     LCT=0
     KDL = 2
     DO 218 ME=2.40
218  FMG(ME) = BL
201  CONTINUE
502  IFL=0
     DO 505 IPAT=1.4
     DO 505 JPAT=1.6
     CONV(JPAT.IPAT)=0.
 505 CONTINUE
     WRITE(NOUT'IREC.2013)
     DO 1301  I=1.NFAC
     LV=LVL(I)
     IF(RAW(I.1.IV).EQ.0 .AND. RAW(I.2.IV).EQ.0) GO TO 301
     IFL=1
     SETUP(1) = CC(I)
     GO TO (302.302.303.304).LV
302  SETUP(2) = 2./(RAW(I.2.IV)-RAW(I.1.IV))
     IF(ABS(SETUP(2)).LT.5.E-5) SETUP(2) = 0.
     SETUP(4) = -SETUP(2)*RAW(I.1.IV) - 1.
     CONV(I.1) = SETUP(2)
     CONV(I.2) = SETUP(4)
     SETUP(3) = AR(I)
     WRITE(NOUT'IREC.92) (SETUP(IB).IB=1.4)
92   FORMAT(1H0.5X.5HWHERE.A3.3H  =.  G14.7.3H * .A4.5H  + ( .
    1  G14.7.1H).11X)
     KPG = KPG + 2
     GO TO 301
303  HI=RAW(I.3.IV)
     AM=RAW(I.2.IV)
     AL=RAW(I.1.IV)
     BET=(2.*AM-HI-AL)/((AM-AL)*(HI-AL)*(HI-AM))
     ALF = 2./(HI-AL)
     SETUP(2) = BET
     SETUP(3)=AR(I)
     SETUP(4) = ALF - BET*(HI+AL)
     SETUP(5) = AR(I)
```

```
      SETUP(6) = HI*AL*BET - 1. -AL*ALF
      IF(ABS(SETUP(2)).LT.5.E-5) SETUP(2) = 0.
      IF(ABS(SETUP(4)).LT.5.E-5) SETUP(4) = 0.
      IF(ABS(SETUP(6)).LT.5.E-5) SETUP(6) = 0.
      CONV(I,1) = SETUP(2)
      CONV(I,2) = SETUP(4)
      CONV(I,3) = SETUP(6)
      WRITE(NOUT'IREC,93) (SETUP(IC),IC=1,6)
93    FORMAT(1H0,5X,5HWHERE,A3,3H  =, G14.7,3H * ,A4,8H**2  + ( ,
     1  G14.7,3H)* ,A4,2X/1H ,5H  + (, G14.7,1H),48X)
      KPG = KPG+2
      GO TO 301
304   VLO=RAW(I,1,IV)
      ALO=RAW(I,2,IV)
      AHI=RAW(I,3,IV)
      VHI=RAW(I,4,IV)
      QA=2./(ALO-VLO)/3.
      QB=4./(AHI-VLO)/3.
      QC=2./(VHI-VLO)
      QB=(QB-QA)/(AHI-ALO)
      QC=(QC-QA)/(VHI-ALO)
      QC=(QC-QB)/(VHI-AHI)
      AN(1) = QC
      AN(2) = QB
      AN(3) = QA
      AN(4) = -1.
      CMP(1) = VLO
      CMP(2)=ALO
      CMP(3) = AHI
      CALL DWMULT (AN,3,CMP,BN)
      DO 324 MG=1,4
      SETUP(2*MG+1) = AR(I)
      SETUP(2*MG) = BN(MG)
      IF(ABS(SETUP(2*MG)).LT.5.E-5) SETUP(2*MG)=0.
324   CONV(I,MG) = SETUP(2*MG)
      WRITE(NOUT'IREC,94) (SETUP(MH),MH=1,8)
94    FORMAT(1H0,2X,5HWHERE,A3,3H  =, G14.7,3H * ,A4,8H**3  + ( ,
     1  G14.7,3H)* ,A4,4H**2 ,1X/4H +( ,  G14.7,3H)* ,A4,5H  + (,  G14.7
     2,1H),24X)
      KPG = KPG+2
301   IF(KPG.GT.48) KPG=0
      KPG = KPG + 1
1301  CONTINUE
      DO 507 IPAT=1,4
      DO 507 JPAT=1,6
      COEF(IPAT,JPAT)=CONV(JPAT,IPAT)
 507  CONTINUE
      WRITE(NOUT'IREC,191) RMS,VNAM(1,IV),VNAM(2,IV)
191   FORMAT(1H0,5X,33HMINIMUM TWO SIGMA ERROR IS + OR - ,F15.5,2X,2A4 ,
     15X)
      WRITE(NOUT'IREC,193) RMSM,VNAM(1,IV),VNAM(2,IV)
193   FORMAT(1H0,5X,33HMAXIMUM TWO SIGMA ERROR IS + OR - ,F15.5,2X,2A4 ,
     15X)
      WRITE(NOUT'IREC,1313)
```

```
1313 FORMAT(' LAST')
99   RETURN
     END
```

```
      SUBROUTINE PAGIT(*,*,*)
      EXTERNAL EOBINT
      DOUBLE PRECISION DSSQ,DER,DUMMY
      DIMENSION TEXT(20),BUS(20)
      DIMENSION LTXT(20),IX(4)
      DIMENSION HEAD1(5),HEAD2(2),HEAD3(3),HEAD4(3)
      DIMENSION INHEAD(4),ABCD(4)
      COMMON /BAKDAT/ A(256),VSE   ,SS(256),NER,SSE,SSER,NERR
      COMMON /YATDAT/ DSSQ,DER,
     3LVL(4),NFG,IRAW,DAT(256),NEND(4,4),TTL(20),
     1VNAM(2),NAME(8),FMT(80),ICODE,IL(6),NIN,NOUT,NDX,NF,
     2NVBL,NREP,CONCOM(4,4),NR(256),FA,FB,ND1(256),IV,NW
      COMMON KEY
      COMMON /DIRAC/ IREC,JPOINT,NDM,LSAV
      DATA HEAD1/'SOUR','CE O','F VA','RIAT','ION '/
      DATA HEAD2/'  SU','M OF'/
      DATA HEAD3/'  SQ','UARE','S   '/
      DATA HEAD4/'MEAN',' SQU','ARES'/
      DATA HEAD5/' F  '/
      DATA INHEAD/'A =','B =','C =','D ='/
      DATA ABCD/'A','B','C','D'/
      DATA BLANK/' '/
      DATA LBLNK /' '/
      DATA HEAD6/' DF '/
      DATA TLAST/'LAST'/
      NC=72
      CALL GRINIT('B')
      CALL GCEOB(EOBINT)
      IREC=1
      IPAGE=1
      KEY=1
    1 CALL NEWPGE
      CALL MASKIT(12)
      CALL STUFF(2,'                                        OPTION TABLE
     1                ')
      CALL STUFF(4,'DEPRESS KEY 1 FOR ANALYSIS OF VARIANCE
     1                ')
      CALL STUFF(6,'DEPRESS KEY 2 FOR RESPONSE EQUATION
     1                ')
      CALL STUFF(8,'DEPRESS KEY 3 FOR SINGLE DEGREE OF FREEDOM SUMS OF S
     1QUARES          ')
      CALL STUFF(10,'DEPRESS KEY 4 TO PLOT RESPONSE SURFACE
     1                ')
      CALL STUFF(12,'DEPRESS KEY 5 TO EDIT RAW DATA
     1                 ')
      CALL SNAP
      CALL TWAIT(1,0,0,0)
      IWHERE=KEY
      GO TO (15,100,300,200,601),IWHERE
   15 CALL NEWPGE
      IREC=1
      CALL MASKIT(13)
      CALL STUFF(41,'DEPRESS KEY 1 FOR SUMS OF SQUARES.
     1                 ')
```

```
      CALL STUFF(42,'DEPRESS KEY 2 FOR MEAN SQUARES.
     1                  ')
      CALL STUFF(43,'DEPRESS KEY 3 FOR F VALUES.
     1                  ')
      CALL STUFF(45,'DEPRESS KEY 30 TO VIEW OPTION TABLE
     1                  ')
      JKEY=KEY
      IF(JKEY.GT.3)JKEY=1
      CALL STUFF(2,'                              ANALYSIS OF VARIANCE
     1                ')
      DO 6 JBL=1,17
      TEXT(JBL)=BLANK
    6 CONTINUE
      DO 7 JSR=1,5
      TEXT(JSR+4)=HEAD1(JSR)
    7 CONTINUE
      TEXT(14)=HEAD6
      GO TO(17,18,19),JKEY
   17 TEXT(15)=HEAD3(1)
      TEXT(16)=HEAD3(2)
      TEXT(17)=HEAD3(3)
      CALL STUFF(4,TEXT)
      DO 16 KSR=1,17
      TEXT(KSR)=BLANK
   16 CONTINUE
      TEXT(15)=HEAD2(1)
      TEXT(16)=HEAD2(2)
      CALL STUFF(3,TEXT)
      GO TO 9
   18 TEXT(15)=HEAD4(1)
      TEXT(16)=HEAD4(2)
      TEXT(17)=HEAD4(3)
      CALL STUFF(4,TEXT)
      GO TO 9
   19 TEXT(16)=HEAD5
      CALL STUFF(4,TEXT)
    9 CONTINUE
      LNCK=0
      DO 10 LINE=5,36
      LNSV=LINE
      READ(NOUT'IREC,1001)(TEXT(I),I=1,13)
      LNCK=LNCK+1
      IF(TEXT(1).EQ.TLAST ) GO TO 30
      READ(NOUT'IREC,2000)(BUS(J),J=1,10)
      LNCK=LNCK+1
      TEXT(14)=BUS(1)
      IWR=3*JKEY-1
      TEXT(15)=BUS(IWR)
      TEXT(16)=BUS(IWR+1)
      TEXT(17)=BUS(IWR+2)
      CALL STUFF(LINE,TEXT)
   10 CONTINUE
   30 CALL SNAP
      CALL TWAIT(1,0,0,0)
```

```
      IREC=IREC-LNCK
      IF(KEY.EQ.30) GO TO 1
      IF(KEY.LE.3) GO TO 15
      GO TO 15
  100 IREC=70
  110 CALL NEWPGE
      CALL MASKIT(14)
      CALL STUFF(42,'DEPRESS KEY 28 TO PAGE BACKWARD.
     1                ')
      CALL STUFF(43,'DEPRESS KEY 29 TO PAGE FORWARD.
     1                ')
      CALL STUFF(45,'DEPRESS KEY 30 TO VIEW OPTION TABLE
     1                ')
      DO 120 LINE=1,40
      KEEP=LINE
      READ(NOUT'IREC,3000)(TEXT(I),I=1,17)
      IF(TEXT(1).EQ.TLAST ) GO TO 130
      KCHK=LINE
      CALL STUFF(LINE,TEXT)
  120 CONTINUE
  130 CALL SNAP
      CALL TWAIT(1,0,0,0)
      IF(KEY.EQ.30) GO TO 1
      IF(KEY.EQ.28)IREC=IREC-KEEP-40
      IF(KEY.EQ.29.AND.KCHK.LT.40) IREC=IREC-KEEP
      IF(IREC.LT.70)IREC=70
      GO TO 110
  300 LOW=1
  310 CALL GERAS(100)
      CALL MASKIT(15)
      CALL SPACE(1)
      REWIND NW
      READ(NW) A,SS,SSE,NER,VSE
      CALL GRDPLY('                          SINGLE DEGREE OF FREEDOM DISPLAY'
     1,52,&1000)
      CALL SPACE(1)
      CALL GRDPLY('DEPRESS KEY 1 TO DELETE TERMS BY CELL NUMBER',44,
     1&1000)
      CALL GRDPLY('DEPRESS KEY 2 TO DELETE TERMS BY F COMPARISON',45,
     1&1000)
      CALL GRDPLY('DEPRESS KEY 28 TO PAGE BACKWARD',31,&1000)
      CALL GRDPLY('DEPRESS KEY 29 TO PAGE FORWARD',30,&1000)
      CALL SPACE(1)
      CALL GRDPLY('DEPRESS KEY 30 TO VIEW OPTION TABLE',35,&1000)
      CALL SPACE(1)
      WRITE(NDM,560) VSE
  560 FORMAT('     MSE = ',G15.8)
      CALL FETCH(TEXT,MC)
      CALL GRDPLY(TEXT,MC,&1000)
      CALL SPACE(1)
      WRITE(NDM,540)(INHEAD(KOUT),NAME(2*KOUT-1),NAME(2*KOUT),KOUT=1,NF)
      CALL FETCH(TEXT,MC)
      CALL GRDPLY(TEXT,MC,&1000)
      CALL SPACE(1)
```

```
 540 FORMAT(4(3A4.4X))
     WRITE(NDM.545) (ABCD(KOUT).KOUT=1.NF)
 545 FORMAT('CELL'.4(1X.A1))
     CALL FETCH(LTXT.MC)
     WRITE(NDM.550) (LTXT(KOUT).KOUT=1.3)
 550 FORMAT(3A4.4X.'COEFFICIENT'.4X.'SUM OF SQUARES'.4X.'F RATIO')
     CALL FETCH(TEXT.MC)
     CALL GRDPLY(TEXT.MC.&1000)
     CALL SPACE(1)
     JHI=LOW+29
     JHI=MINO(JHI.NFG)
     DO 600 J=LOW.JHI
     CALL ANAL(J.IX(1).IX(2).IX(3).IX(4))
     DO 1600 ISUB=1.4
     IX(ISUB)=IX(ISUB)-1
1600 CONTINUE
     F=SS(J)/VSE
     LTXT(2)=LBLNK
     LTXT(3)=LBLNK
     WRITE(NDM.570) J.(IX(KOUT).KOUT=1.NF)
     CALL FETCH(LTXT.MC)
 570 FORMAT(I2,2X,4I2)
     WRITE(NDM.580)(LTXT(KOUT).KOUT=1.3).A(J).SS(J).F
 580 FORMAT(3A4.2X.G15.8.2X.G15.8.3X.G15.8)
     CALL FETCH(TEXT.MC)
     CALL GRDPLY(TEXT.MC.&1000)
 600 CONTINUE
     CALL TWAIT(1.0.0.0)
     CALL GERAS(100)
     IF(KEY.EQ.1) GO TO 400
     IF(KEY.EQ.2) GO TO 500
     IF(KEY.EQ.30) GO TO 1
     IF(KEY.EQ.28) LOW=MAXO(1.LOW-30)
     IF(KEY.EQ.29) LOW=LOW+30
     IF(LOW.GT.NFG) LOW=LOW-30
     GO TO 310
 400 CALL SPACE(1)
     CALL MASKIT(1)
     CALL GRDPLY('IF YOU WISH THE FOLLOWING CHANGES TO BE MADE PERMANEN
    1T. DEPRESS KEY 1'.69.&1000)
     CALL GRDPLY('OTHERWISE. DEPRESS KEY 2'.24.&1000)
     CALL TWAIT(1.0.0.0)
     CALL GERAS(100)
     IPERM=KEY
     CALL SPACE(1)
     CALL GRDPLY('ENTER. SEPARATED BY COMMAS. THE CELLS FOR WHICH YOU W
    1ISH TERMS DELETED'.70.&1000)
 409 CALL TWAIT(0.0.1.0)
     CALL GRRPLX(TEXT.NC)
     INDEX=0
 410 CALL INX(TEXT.INDEX.NC.DUMMY.&1100)
     KP=DUMMY+.05
     A(KP)=0.
     SSE=SSE+SS(KP)
```

```
      NER=NER+1
      IF(INDEX.GT.0) GO TO 410
  420 CONTINUE
      VSE=SSE/FLOAT(NER)
      IF(IPERM.EQ.2) GO TO 305
      REWIND NW
      WRITE(NW) A.SS.SSE.NER.VSE
  305 CALL GRRLSE
      RETURN 3
  500 CALL SPACE(1)
      CALL MASKIT(1)
      CALL GRDPLY('IF YOU WISH THE FOLLOWING CHANGES TO BE MADE PERMANEN
     1T. DEPRESS KEY 1',69,&1000)
      CALL GRDPLY('OTHERWISE. DEPRESS KEY 2'.24.&1000)
      CALL TWAIT(1.0.0.0)
      CALL GERAS(100)
      IPERM=KEY
      CALL SPACE(1)
      CALL GRDPLY('ENTER THE F VALUE'.17.&1000)
  408 CALL TWAIT(0.0.1.0)
      CALL GRRPLX(TEXT.NC)
      INDEX=0
      CALL INX(TEXT.INDEX.NC.DUMMY.&1200)
      FA=DUMMY
      DO 162 KP=1.NFC
      IF(SS(KP)/VSE.GE.FA) GO TO 162
      A(KP)=0.
      SSE=SSE+SS(KP)
      NER=NER+1
  162 CONTINUE
      GO TO 420
 1000 CONTINUE
  200 CALL GRRLSE
      RETURN 2
 1100 CALL GRDPLY('AN INVALID NUMBER WAS ENTERED. TRY AGAIN'.40.&1000)
      GO TO 409
 1200 CALL GRDPLY('AN INVALID NUMBER WAS ENTERED. TRY AGAIN'.40.&1000)
      GO TO 408
  601 CALL GRRLSE
      RETURN 1
 1001 FORMAT(1X.13A4)
 2000 FORMAT(1X.10A4)
 3000 FORMAT(1X.17A4)
      END
```

```
      SUBROUTINE PLOTTR(*,*)
      EXTERNAL EOBINT
      COMMON KEY
    1 CALL GRINIT('B')
      CALL OCEOB(EOBINT)
      CALL MASKIT(12)
      CALL NEWPGE
      CALL STUFF(2,'                              OPTION TABLE
     1                ')
      CALL STUFF(4,'DEPRESS KEY 1 TO PLOT THE PREDICTED RESPONSE AGAINST
     1 ONE FACTOR      ')
      CALL STUFF(6,'DEPRESS KEY 2 TO PLOT A 95% CONFIDENCE INTERVAL AGAI
     1NST ONE FACTOR  ')
      CALL STUFF(8,'DEPRESS KEY 3 TO PLOT CONTOURS
     1                ')
      CALL STUFF(10,'DEPRESS KEY 4 TO EDIT THE RAW DATA
     1                ')
      CALL STUFF(12,'DEPRESS KEY 5 TO RETURN TO THE RESPONSE EQUATION
     1                ')
      CALL SNAP
      CALL TWAIT(1,0,0,0)
      IWHERE=KEY
      CALL GERAS(100)
      CALL GRRLSE
      GO TO (10,20,30,40,50),IWHERE
   50 RETURN 2
   40 RETURN 1
   30 CALL CONTUR
      GO TO 1
   20 CALL CONF
      GO TO 1
   10 CALL SPLOT
      GO TO 1
      END
```

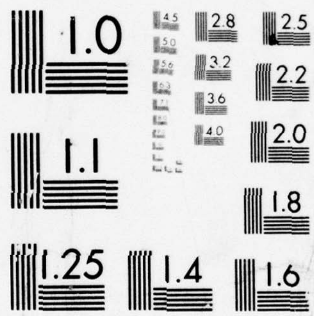
```
      SUBROUTINE SPLOT
      EXTERNAL EOBINT
      DOUBLE PRECISION DUMMY
      DOUBLE PRECISION DSSQ.DER
      REAL NAME
      COMMON KEY
      COMMON /YATDAT/ DSSQ.DER.
     3LVL(4).NFG.IRAW.DAT(256).NEND(4.4).TTL(20).
     1VNAM(2).NAME( 8).FMT(80).ICODE.IL(6).        NIN.NOUT.NDX.NF.
     2NVBL.NREP.CONCOM(4.4).NR(256).FA.FB.ND1(256).IV.NW.COEF(4.6).
     4A(512)
      DIMENSION TEXT(20).ALIST(6).PLOFAC(5).BUS(20).RAWVAR(6.4).
     1INDVAR(6).NORDR(6).PLOTNO(2).VAL(6).PLIST(4).C(512).IX(6)
      DIMENSION CTEMP(512)
      DATA ALIST/' 1. '.' 2. '.' 3. '.' 4. '.' 5. '.' 6. '/
      DATA BLANK/' '/
      DATA PLOFAC/' **P'.'LOTT'.'ING '.'FACT'.'OR**'/
      DATA PLOTNO/'PLOT'.' NO.'/
      DATA PLIST/'1'.'2'.'3'.'4'/
 1000 FORMAT('PLEASE ENTER THE VALUE(S) FOR '.2A4.2X)
 1001 FORMAT(2X.G10.3)
 1002 FORMAT(G10.3.2X)
 1003 FORMAT('INCREMENT='.G10.3)
 1004 FORMAT('INCREMENT=')
 1010 FORMAT(20X.4G10.3)
      WRITE(6.1010) COEF
      NS=68
      NDM=4
      CALL INITP
      CALL PASKIT(2)
      CALL EOBP(EOBINT)
   45 CALL PEWPGE
      CALL GRAXES(0..4095..80..4095..0.4095.80.4095)
      DO 17 KOB=1.6
      INDVAR(KOB)=1
      NORDR(KOB)=KOB
   17 CONTINUE
      CALL PSTUFF(2.' THE FACTORS ARE AS FOLLOWS:
     1                 ')
      DO 40 I=4.17
      TEXT(I)=BLANK
   40 CONTINUE
      DO 50 I=1.NF
      J=2*I-1
      TEXT(1)=ALIST(I)
      TEXT(2)=NAME(J)
      TEXT(3)=NAME(J+1)
      CALL PSTUFF(I+3.TEXT)
   50 CONTINUE
      CALL PSTUFF(11.'TYPE IN THE NUMBER OF THE FACTOR TO BE PLOTTED.
     1                 ')
      CALL PSNAP
      CALL PWAIT(0.0.1.0)
      CALL GREPLX(TEXT.NS)
```

AD-A036 346 GEORGIA UNIV ATHENS DEPT OF STATISTICS AND COMPUTER--ETC F/G 9/2
GRAPHICAL AIDS FOR STATISTICAL COMPUTATION.(U)
DEC 76 W P BOND, R E BARGMANN N00014-69-A-0423

UNCLASSIFIED TR-112 NL

4 OF 4
AD
A036346

END
DATE
FILMED
3-77



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
2000 FORMAT(20X,'TEXT=',20A4)
     INDEX=0
     CALL INX(TEXT,INDEX,NS,DUMMY,&900)
     IVAR=DUMMY+.01
     ILN=IVAR+3
     CALL PPLUCK(ILN,TEXT)
     DO 55 I=1,5
     TEXT(I+3)=PLOFAC(I)
  55 CONTINUE
     CALL PSTUFF(ILN,TEXT)
 157 CALL PSTUFF(11,'ENTER SEPARATED BY A COMMA THE MINIMUM AND MAXIMUM
    1                    ')
     CALL PSTUFF(12,'VALUE TO BE PLOTTED.
    1                    ')
     CALL PSNAP
     CALL PWAIT(0.0,1.0)
     CALL GREPLX(TEXT,NS)
     INDEX=0
     CALL INX(TEXT,INDEX,NS,DUMMY,&900)
     VARMIN=DUMMY
     CALL INX(TEXT,INDEX,NS,DUMMY,&900)
     VARMAX=DUMMY
     IF(VARMIN.LT.VARMAX) GO TO 56
     TEMP=VARMIN
     VARMIN=VARMAX
     VARMAX=TEMP
  56 CALL PPLUCK(ILN,BUS)
     DO 156 IPUT=1,16
     BUS(IPUT+3)=TEXT(IPUT)
 156 CONTINUE
     CALL PSTUFF(ILN,BUS)
     LVLUP=LVL(IVAR)
     IF(VARMIN.LT.CONCOM(IVAR,1).OR.VARMIN.GT.CONCOM(IVAR,LVLUP))
    1 GO TO 158
     IF(VARMAX.LT.CONCOM(IVAR,1).OR.VARMAX.GT.CONCOM(IVAR,LVLUP))
    1 GO TO 158
     GO TO 168
 158 CALL PSTUFF(13,'YOU ARE EXTRAPOLATING
    1                    ')
     CALL PSTUFF(14,'DEPRESS KEY 1 TO CONTINUE
    1                    ')
     CALL PSTUFF(15,'DEPRESS KEY 2 TO RE-ENTER VALUES
    1                    ')
     CALL PSNAP
     CALL PWAIT(1.0,0.0)
     CALL PEWLNE(13)
     CALL PEWLNE(14)
     CALL PEWLNE(15)
     IF(KEY.EQ.2) GO TO 157
 168 CALL PSTUFF(13,'UP TO FOUR PLOTS CAN BE VIEWED SIMULTANEOUSLY.
    1                    ')
     CALL PSTUFF(14,'TO ACCOMPLISH THIS, SIMPLY TYPE MORE THAN ONE VALU
    1E.                  ')
     CALL PSTUFF(15,'(SEPARATED BY COMMAS) WHEN ASKED TO ENTER THE FOLL
```

```
     1OWING.                 ')
      CALL PSTUFF(16,'FIXED VALUES.
     1                       ')
      CALL PSTUFF(17,'EXAMPLE:
     1                       ')
      CALL PSTUFF(18,'10.5,15.2
     1                       ')
      CALL PSNAP
      DO 60 I=1,NF
      J=2*I-1
      IF(I.EQ.IVAR) GO TO 60
   92 IXTR=0
      WRITE(4,1000) NAME(J),NAME(J+1)
      CALL FETCH(TEXT,MS)
      CALL PSTUFF(21,TEXT)
      CALL PSNAP
      CALL PWAIT(0,0,1,0)
      CALL GREPLX(BUS,NS)
      CALL PPLUCK(I+3,TEXT)
      DO 57 K=1,16
      TEXT(K+3)=BUS(K)
   57 CONTINUE
      CALL PSTUFF(I+3,TEXT)
      INDVAR(I)=0
      INDEX=0
   59 IF(INDVAR(I).GT.4) GO TO 60
      CALL INX(BUS,INDEX,NS,DUMMY,&960)
      INDVAR(I)=INDVAR(I)+1
      JMUL=INDVAR(I)
      RAWVAR(I,JMUL)=DUMMY
      LVLUP=LVL(I)
      IF(DUMMY.LT.CONCOM(I,1).OR.DUMMY.GT.CONCOM(I,LVLUP)) IXTR=1
      IF(INDEX.GT.0) GO TO 59
      IF(IXTR.EQ.0) GO TO 60
      CALL PSTUFF(I+4,'YOU ARE EXTRAPOLATING
     1                         ')
      CALL PSTUFF(I+5,'DEPRESS KEY 1 TO CONTINUE
     1                         ')
      CALL PSTUFF(I+6,'DEPRESS KEY 2 TO RE-ENTER VALUES
     1                         ')
      CALL PSNAP
      CALL PWAIT(1,0,0,0)
      CALL PEWLNE(I+4)
      CALL PEWLNE(I+5)
      CALL PEWLNE(I+6)
      IF(KEY.EQ.2) GO TO 92
   60 CONTINUE
      WRITE(6,1050) RAWVAR
      WRITE(6,1060) INDVAR
 1050 FORMAT(20X,6F6.2)
 1060 FORMAT(20X,6I3)
      KNTPLO=1
      DO 65 I=1,NF
      IF(I.EQ.IVAR) GO TO 65
```

```
      KNTPLO=KNTPLO*INDVAR(I)
   65 CONTINUE
      IF(KNTPLO.LE.4) GO TO 75
      CALL PSTUFF(23.'YOU HAVE REQUESTED MORE THAN FOUR PLOTS.
     1                  ')
      CALL PSTUFF(24.'DEPRESS KEY 1 TO RESTART THIS SECTION.
     1                  ')
      CALL PSNAP
      CALL PWAIT(1.0.0.0)
      GO TO 45
   75 CONTINUE
      CALL PSTUFF(23.'DEPRESS KEY 1 TO PLOT.
     1                  ')
      CALL PSNAP
      CALL PWAIT(1.0.0.0)
      CALL REFILL
      TEXT(1)=PLOTNO(1)
      TEXT(2)=PLOTNO(2)
      KKK=3
      DO 190 JJ=1.NF
      IF(JJ.EQ.IVAR) GO TO 190
      KK=2*JJ-1
      TEXT(KKK)=BLANK
      TEXT(KKK+1)=NAME(KK)
      TEXT(KKK+2) =NAME(KK+1)
      KKK=KKK+3
  190 CONTINUE
      NCHAR=12*NF-4
      TOP=4055.
      CALL GRCHAR('BP'.TEXT.NCHAR.23..TOP.IER)
      NCHAR=NCHAR+2
      JTLVL=LVL(IVAR)-1
      RNG=VARMAX-VARMIN
      DEL=RNG/50.
      DELLBL=RNG/4.
      NX=55*KNTPLO
      CALL UVOO(NX)
      CALL XCHNGE(IVAR.CTEMP)
      WRITE(6.1020) (A(JOUT).CTEMP(JOUT).JOUT=1.NFG)
 1020 FORMAT(20X.2E12.5)
      NORDR(IVAR)=NF
      NORDR(NF)=IVAR
      DO 300 IPLOT=1.KNTPLO
      CALL BNAL(IPLOT.IX(1).IX(2).IX(3).IX(4).INDVAR)
      MF=NF-1
      DO 200 J=1.MF
      JT=NORDR(J)
      JCOL=IX(JT)
      JTLVL=LVL(JT)-1
      VAL(J)=POLY(COEF(1.JT).RAWVAR(JT.JCOL).JTLVL)
  200 CONTINUE
      WRITE(6.1050) VAL
      TEXT(1)=ALIST(IPLOT)
      TEXT(2)=BLANK
```

```
      KK=3
      DO 205 J=1.NF
      IF(J.EQ.IVAR) GO TO 205
      JCOL=IX(J)
      WRITE(NDM.1001) RAWVAR(J    .JCOL)
      CALL FETCH(BUS.MC)
      TEXT(KK)=BUS(1)
      TEXT(KK+1)=BUS(2)
      TEXT(KK+2)=BUS(3)
      KK=KK+3
  205 CONTINUE
      TOP=TOP-80.
      CALL GRCHAR('BP'.TEXT.NCHAR.23..TOP.IER)
      CALL SPOLY(CTEMP.VAL.MF.NORDR.C)
      JTLVL=LVL(IVAR)
      RMIN=10.E30
      RMAX=-RMIN
      XVAR=VARMIN
      DO 210 I=1.3
      XVAR=XVAR+DELLBL
      XVAL=POLY(COEF(1.IVAR).XVAR.LVL(IVAR)-1)
      R=BPOLY(C.XVAL.JTLVL)
      CALL PUTUV(XVAR.R.KERROR)
  210 CONTINUE
      XVAR=VARMIN
      DO 220 J=1.51
      XVAL= POLY(COEF(1.IVAR).XVAR.LVL(IVAR)-1)
      R=BPOLY(C.XVAL.JTLVL)
      CALL PUTUV(XVAR.R.KERROR)
      IF(R.LT.RMIN) RMIN=R
      IF(R.GT.RMAX) RMAX=R
      XVAR=XVAR+DEL
  220 CONTINUE
      IF(IPLOT.EQ.1) GO TO 290
      IF(RMIN.LT.SMIN) SMIN=RMIN
      IF(RMAX.GT.SMAX) SMAX=RMAX
      GO TO 300
  290 SMIN=RMIN
      SMAX=RMAX
  300 CONTINUE
      WRITE(NDM.1002) VARMIN
      CALL FETCH(BUS.MC)
      CALL GRCHAR('BP'.BUS.MC.364..460..IER)
      WRITE(NDM.1002) VARMAX
      CALL FETCH(BUS.MC)
      CALL GRCHAR('BP'.BUS.MC.3364..460..IER)
      VARINC=RNG/10.
      WRITE(NDM.1003) VARINC
      CALL FETCH(BUS.MC)
      CALL GRCHAR('BP'.BUS.MC.1640..440..IER)
      BUS(1)=NAME(2*IVAR-1)
      BUS(2)=NAME(2*IVAR)
      CALL GRCHAR('BP'.BUS.8.1976..520..IER)
      RESRNG=SMAX-SMIN
```

```
      RINC=RESRNG/10.
      WRITE(NDM.1002) SMIN
      CALL FETCH(BUS.MC)
      CALL GRCHAR('BP'.BUS.MC.0..560..IER)
      WRITE(NDM.1002) SMAX
      CALL FETCH(BUS.MC)
      CALL GRCHAR('BP'.BUS.MC.0..3560..IER)
      WRITE(NDM.1004)
      CALL FETCH(BUS.MC)
      CALL GRCHAR('BP'.BUS.MC.0..2140..IER)
      WRITE(NDM.1002) RINC
      CALL FETCH(BUS.MC)
      CALL GRCHAR('BP'.BUS.MC.0..2060..IER)
      CALL GRCHAR('BP'.'DEPRESS KEY 1 TO RETURN TO OPTION TABLE'.
     1 39.500..340..IER)
      CALL GRAXES(VARMIN.VARMAX.SMIN.SMAX.700.3700.600.3600)
      CALL GRGRID(RNG.RESRNG.'A'.IFLAG)
      XLBL=VARMIN
      YLBL=SMIN
      DO 512 LBL=1.9
      XLBL=XLBL+VARINC
      YLBL=YLBL+RINC
      CALL GRCHAR('BP'.'\'.1.XLBL.SMIN.IER)
      CALL GRCHAR('BP'.'-'.1.VARMIN.YLBL.IER)
  512 CONTINUE
      DO 400 J=1.KNTPLO
      JJ=(J-1)*54+1
      CALL GRPLOT('BP'.'B'.'A'.JJ.3.PLIST(J).'O'.IER)
      JJP=JJ+3
      CALL GRPLOT('BP'.'A'.'A'.JJP.51.'*'.'O'.IER)
  400 CONTINUE
      CALL PWAIT(1.0.0.0)
      CALL UV99
      CALL REFILL
      CALL RLSEP
      RETURN
  900 CONTINUE
  960 CONTINUE
      WRITE(6.2001)
 2001 FORMAT(20X.'BAD CONVERSION')
      CALL RLSEP
      RETURN
      END
```

```
      SUBROUTINE CONF
      EXTERNAL EOBINT
      DOUBLE PRECISION DUMMY.DSSQ.DER
      REAL NAME
      COMMON KEY
      COMMON /YATDAT/ DSSQ.DER.
     3LVL(4).NFG.IRAW.DAT(256).NEND(4.4).TTL(20).
     1VNAM(2).NAME(8).FMT(80).ICODE.IL(6).NIN.NOUT.NDX.NF.
     2NVBL.NREP.CONCOM(4.4).NR(256).FA.FB.ND1(256).IV.NW.COEF(4.6).
     4A(512)
      COMMON /BAKDAT/ AFILL(256),VSE.BFIL(260)
      COMMON /DIRAC/ IREC.JPOINT.NDM.LSAV
      DIMENSION TEXT(20).ALIST(6).PLOFAC(5).BUS(20).RAWVAR(4).
     1INDVAR(6).NORDR(6).VAL(6).ADD(4)   .C(256).IX(6).
     2CTEMP(256).XUP(51).XLO(51).PHI(4.4).LVT(4)
      DATA BLANK/' '/
      DATA PLOFAC/' **P','LOTT','ING ','FACT','OR**'/
      DATA ALIST /' 1. ',' 2. ',' 3. ',' 4. '/
 1001 FORMAT(2X.G10.3)
 1002 FORMAT(G10.3.2X)
 1003 FORMAT('INCREMENT=',G10.3)
 1004 FORMAT('INCREMENT=')
      WRITE(6.1010) COEF
 1010 FORMAT(20X.4G10.3)
      NS=68
      CALL INITP
      CALL EOBP(EOBINT)
      CALL PEWPGE
      CALL GRAXES(0..4095..80..4095..0.4095.80.4095)
      DO 17 KOB=1.4
      NORDR(KOB)=KOB
   17 CONTINUE
      CALL PASKIT(2)
      CALL PSTUFF(2.'THE FACTORS ARE AS FOLLOWS:
     1                 ')
      DO 40 I=1.17
      TEXT(I)=BLANK
   40 CONTINUE
      DO 50 I=1.NF
      J=2*I-1
      TEXT(1)=ALIST(I)
      TEXT(2)=NAME(J)
      TEXT(3)=NAME(J+1)
      CALL PSTUFF(I+3.TEXT)
   50 CONTINUE
      CALL PSTUFF(11.'TYPE IN THE NUMBER OF THE FACTOR TO BE PLOTTED.
     1                 ')
      CALL PSNAP
      CALL PWAIT(0.0.1.0)
      CALL GREPLX(TEXT.NS)
      INDEX=0
      CALL INX(TEXT.INDEX.NS.DUMMY.&900)
      IVAR=DUMMY+.1
      ILN=IVAR+3
```

```
      CALL PPLUCK(ILN,TEXT)
      DO 55 I=1,5
      TEXT(I+3)=PLOFAC(I)
   55 CONTINUE
      CALL PSTUFF(ILN,TEXT)
      CALL PSTUFF(11,'ENTER SEPARATED BY A COMMA THE MINIMUM AND MAXIMUM
     1                    ')
      CALL PSTUFF(12,'VALUE TO BE PLOTTED.
     1                    ')
   54 CALL PSNAP
      CALL PWAIT(0.0,1.0)
      CALL GREPLX(TEXT,NS)
      INDEX=0
      CALL INX(TEXT,INDEX,NS,DUMMY,&900)
      VARMIN=DUMMY
      CALL INX(TEXT,INDEX,NS,DUMMY,&900)
      VARMAX=DUMMY
      IF(VARMIN.LT.VARMAX) GO TO 56
      TEMP=VARMIN
      VARMIN=VARMAX
      VARMAX=TEMP
   56 IF(VARMIN.GE.CONCOM(IVAR,1).AND.VARMAX.LE.CONCOM(IVAR,LVL(IVAR)))
     1 GO TO 57
      CALL PSTUFF(14,'VALUES OUTSIDE OF THE RANGE HAVE BEEN CHOSEN.
     1                    ')
      CALL PSTUFF(15,'DEPRESS KEY 1 TO CONTINUE OR KEY 2 TO RESELECT THE
     1 VALUES             ')
      CALL PSNAP
      CALL PWAIT(1.0,0.0)
      CALL PEWLNE(14)
      CALL PEWLNE(15)
      IF(KEY.EQ.2) GO TO 54
   57 CALL PPLUCK(ILN,BUS)
      DO 156 IPUT=1,16
      BUS(IPUT+3)=TEXT(IPUT)
  156 CONTINUE
      CALL PSTUFF(ILN,BUS)
      CALL PSNAP
      DO 60 I=1,NF
      J=2*I-1
      IF(I.EQ.IVAR) GO TO 60
      WRITE(NDM,1000) NAME(J),NAME(J+1)
 1000 FORMAT('ENTER THE VALUE FOR ',2A4)
      CALL FETCH(TEXT,MS)
      CALL PSTUFF(14,TEXT)
   59 CALL PSNAP
      CALL PWAIT(0.0,1.0)
      CALL GREPLX(BUS,NS)
      CALL PPLUCK(I+3,TEXT)
      DO 157 K=1,16
      TEXT(K+3)=BUS(K)
  157 CONTINUE
      CALL PSTUFF(I+3,TEXT)
      INDEX=0
```

```
      CALL INX(BUS,INDEX,NS,DUMMY,&960)
      RAWVAR(I)=DUMMY
      IF(RAWVAR(I).GE.CONCOM(I,1).AND.RAWVAR(I).LE.CONCOM(I,LVL(I)))
     1 GO TO 60
      CALL PSTUFF(15,'THE VALUE IS OUTSIDE OF THE RANGE.
     1               ')
      CALL PSTUFF(16,'DEPRESS KEY 1 TO CONTINUE OR KEY 2 TO RESELECT THE
     1 VALUE.          ')
      CALL PSNAP
      CALL PWAIT(1,0,0,0)
      CALL PEWLNE(15)
      CALL PEWLNE(16)
      IF(KEY.EQ.2) GO TO 59
   60 CONTINUE
      WRITE(6,1050) RAWVAR
      WRITE(6,1060) INDVAR
 1050 FORMAT(20X,6F6.2)
 1060 FORMAT(20X,6I3)
      CALL PSTUFF(14,'DEPRESS KEY 1 TO PLOT
     1               ')
      CALL PSNAP
      CALL PWAIT(1,0,0,0)
      CALL REFILL
      TEXT(1)=BLANK
      TEXT(2)=BLANK
      KKK=3
      DO 190 JJ=1,NF
      IF(JJ.EQ.IVAR) GO TO 190
      KK=2*JJ-1
      TEXT(KKK)=BLANK
      TEXT(KKK+1) = NAME(KK)
      TEXT(KKK+2)=NAME(KK+1)
      KKK=KKK+3
  190 CONTINUE
      NCHAR=12*NF-4
      TOP=4055.
      CALL GRCHAR('BP',TEXT,NCHAR,23.,TOP,IER)
      NCHAR=NCHAR+2
      JTLVL=LVL(IVAR)+2
      RNG=VARMAX-VARMIN
      DEL=RNG/50.
      NX=165
      CALL UVOO(NX)
      CALL XCHNGE(IVAR,CTEMP)
      WRITE(6,1020)(A(JOUT),CTEMP(JOUT),JOUT=1,NFG)
 1020 FORMAT(20X,2E12.5)
      NORDR(IVAR)=NF
      NORDR(NF)=IVAR
      MF=NF-1
      DO 200 J=1,MF
      JT=NORDR(J)
      JTLVL=LVL(JT)-1
      VAL(J)=POLY(COEF(1,JT),RAWVAR(JT),JTLVL)
  200 CONTINUE
```

```
      WRITE(6,1050) VAL
      TEXT(1)=BLANK
      TEXT(2)=BLANK
      KK=3
      DO 205 J=1,NF
      IF(J.EQ.IVAR) GO TO 205
      WRITE(NDM,1001) RAWVAR(J)
      CALL FETCH(BUS,MC)
      TEXT(KK)=BUS(1)
      TEXT(KK+1)=BUS(2)
      TEXT(KK+2)=BUS(3)
      KK=KK+3
  205 CONTINUE
      TOP=TOP-80.
      CALL ORCHAR('BP',TEXT,NCHAR,23.,TOP,IER)
      DO 690 KK=1,4
  690 ADD(KK)=0.
      ANT=NFG*NREP
      AN=1.+1.0/ANT
      DO 700 J=1,MF
      JLVL=LVL(NORDR(J))
      DO 700 K=1,JLVL
      PHI(K,J)=ORTH(VAL(J),K,JLVL)
  700 CONTINUE
      DO 800 K=1,NFG
      CALL ANAL(K,IX(1),IX(2),IX(3),IX(4))
      AD=ND1(K)
      ANUM=1.0
      DO 990 I=1,MF
      II=NORDR(I)
      ANUM=ANUM*PHI(IX(II),I)
  990 CONTINUE
      JTEMP=IX(IVAR)
      ADD(JTEMP)=ADD(JTEMP)+ANUM*ANUM/AD
  800 CONTINUE
      CALL SPOLY(CTEMP,VAL,MF,NORDR,C)
      JTLVL=LVL(IVAR)
      SMIN=10.E30
      SMAX=-SMIN
      XVAR=VARMIN
      DO 220 J=1,51
      XVAL=POLY(COEF(1,IVAR),XVAR,JTLVL-1)
      R=BPOLY(C,XVAL,JTLVL)
      CALL PUTUV(XVAR,R,KERROR)
      ANN=AN
      DO 225 JJ=1,JTLVL
      FEE=ORTH(XVAL,JJ,JTLVL)
      ANN=ANN+ADD(JJ)*FEE*FEE
  225 CONTINUE
      TERM=2.0*SQRT(VSE*ANN)
      XLO(J)=R-TERM
      IF(XLO(J).LT.SMIN) SMIN=XLO(J)
      XUP(J)=R+TERM
      IF(XUP(J).GT.SMAX) SMAX=XUP(J)
```

```
      XVAR=XVAR+DEL
  220 CONTINUE
      XVAR=VARMIN
      DO 230 J=1,51
      CALL PUTUV(XVAR,XLO(J),KERROR)
      XVAR=XVAR+DEL
  230 CONTINUE
      XVAR=VARMIN
      DO 240 K=1,51
      CALL PUTUV(XVAR,XUP(K),KERROR)
      XVAR=XVAR+DEL
  240 CONTINUE
      WRITE(NDM,1002) VARMIN
      CALL FETCH(BUS,MC)
      CALL GRCHAR('BP',BUS,MC,364.,460.,IER)
      WRITE(NDM,1002) VARMAX
      CALL FETCH(BUS,MC)
      CALL GRCHAR('BP',BUS,MC,3364.,460.,IER)
      VARINC=RNG/10.
      WRITE(NDM,1003) VARINC
      CALL FETCH(BUS,MC)
      CALL GRCHAR('BP',BUS,MC,1640.,440.,IER)
      BUG(1)=NAME(2*IVAR-1)
      BUG(2)=NAME(2*IVAR)
      CALL GRCHAR('BP',BUG,8,1976.,520.,IER)
      RESRNG=SMAX-SMIN
      RINC=RESRNG/10.
      WRITE(NDM,1002) SMIN
      CALL FETCH(BUS,MC)
      CALL GRCHAR('BP',BUS,MC,0.,560.,IER)
      WRITE(NDM,1002) SMAX
      CALL FETCH(BUS,MC)
      CALL GRCHAR('BP',BUS,MC,0.,3560.,IER)
      WRITE(NDM,1004)
      CALL FETCH(BUS,MC)
      CALL GRCHAR('BP',BUS,MC,0.,2140.,IER)
      WRITE(NDM,1002) RINC
      CALL FETCH(BUS,MC)
      CALL GRCHAR('BP',BUS,MC,0.,2060.,IER)
      CALL GRCHAR('BP','DEPRESS KEY 1 TO RETURN TO OPTION TABLE',
     1 39,500.,340.,IER)
      CALL GRAXES(VARMIN,VARMAX,SMIN,SMAX,700,3700,600,3600)
      CALL GRGRID(RNG,RESRNG,'A',IFLAG)
      XLBL=VARMIN
      YLBL=SMIN
      DO 512 LBL=1,9
      XLBL=XLBL+VARINC
      YLBL=YLBL+RINC
      CALL GRCHAR('BP','\',1,XLBL,SMIN,IER)
      CALL GRCHAR('BP','-',1,VARMIN,YLBL,IER)
  512 CONTINUE
      CALL GRPLOT('BP','A','A',1,51,'*','0',IER)
      CALL GRPLOT('BP','B','A',52,51,'L','0',IER)
      CALL GRPLOT('BP','B','A',103,51,'U','0',IER)
```

```
      CALL PWAIT(1,0,0,0)
      CALL UV99
      CALL REFILL
      CALL RLSEP
      RETURN
  900 CONTINUE
  960 CONTINUE
      WRITE(6,2001)
 2001 FORMAT(20X,'BAD CONVERSION')
      CALL RLSEP
      RETURN
      END
```

```
      SUBROUTINE CONTUR
      EXTERNAL EOBINT
      DOUBLE PRECISION DUMMY.DSSQ.DER
      REAL NAME
      COMMON KEY.IOVLY.ITYPE
      COMMON /YATDAT/ DSSQ.DER,
     3LVL(4).NFG.IRAW.DAT(256).NEND(4.4).TTL(20).
     1VNAM(2).NAME(8).FMT(80).ICODE.IL(6).NIN.NOUT.NDX.NF.
     2NVBL.NREP.CONCOM(4.4).NR(256).FA.FB.ND1(256).IV.NW.COEF(4.6).
     4A(256)
      COMMON /DIRAC/ IREC.JPOINT.NDM.LSAV
      DIMENSION TEXT(20).ALIST(6).PLOFAC(5).BUS(20).RAWVAR(4).
     1INDVAR(4).NORDR(4).PLOTNO(2).VAL(6).PLIST(4).C(256).IX(6).
     2 CINV(4).BAD(3).JX(4).B(6).IVAR(2).VARMIN(2).VARMAX(2).LVT(4).
     3 ROOT(4).VARINC(2)
      DATA BAD /'NO S'.'OLUT'.'ION '/
      DATA ALIST/' 1. ',' 2. ',' 3. ',' 4. '/
      DATA BLANK/' '/
      DATA PLOFAC/' **P'.'LOTT'.'ING '.'FACT'.'OR**'/
      DATA PLOTNO/'PLOT'.' NO.'/
      DATA PLIST/'1'.'2'.'3'.'4'/
      DATA RESPAL .ONSE/'RESP'.'ONSE'/
 1000 FORMAT('ENTER THE VALUE FOR '.2A4.40X)
 1001 FORMAT(G12.6)
 1002 FORMAT(G10.3.2X)
 1003 FORMAT('INCREMENT='.G10.3)
 1004 FORMAT('INCREMENT=')
 1010 FORMAT(20X.4G10.3)
      NS=68
      CALL INITP
      CALL EOBP(EOBINT)
      CALL PEWPGE
      CALL GRAXES(0..4095..80..4095..0.4095.80.4095)
      CALL PASKIT(2)
      CALL PSTUFF(2.'THE FACTORS ARE AS FOLLOWS:
     1                    ')
      DO 40 I=1.17
      TEXT(I)=BLANK
   40 CONTINUE
      DO 50 I=1.NF
      J=2*I-1
      TEXT(1)=ALIST(I)
      TEXT(2)=NAME(J)
      TEXT(3)=NAME(J+1)
      CALL PSTUFF(I+3.TEXT)
   50 CONTINUE
      CALL PSTUFF(11.'TYPE IN. SEPARATED BY A COMMA. THE NUMBERS OF THE
     1                     ')
      CALL PSTUFF(12.'TWO FACTORS FOR WHICH YOU WISH CONTOURS.
     1                     ')
      CALL PSNAP
      CALL PWAIT(0.0.1.0)
      CALL GREPLX(TEXT.NS)
      INDEX=0
```

```
      CALL INX(TEXT.INDEX.NS.DUMMY.&900)
      IVAR(1)=DUMMY+.1
      CALL INX(TEXT.INDEX.NS.DUMMY.&900)
      IVAR(2)=DUMMY+.1
      IF(LVL(IVAR(2)).LE.LVL(IVAR(1))) GO TO 51
      ITEMP=IVAR(1)
      IVAR(1)=IVAR(2)
      IVAR(2)=ITEMP
   51 CALL ZTORAW(CINV.IVAR(2))
      DO 155 KK=1.2
      ILN=IVAR(KK)+3
      CALL PPLUCK(ILN.TEXT)
      DO 55 I=1.5
      TEXT(I+3)=PLOFAC(I)
   55 CONTINUE
      CALL PSTUFF(ILN.TEXT)
      CALL PSTUFF(11.'ENTER. SEPARATED BY A COMMA. THE MINIMUM AND MAXIM
     1UM                ')
      CALL PSTUFF(12.'VALUE FOR THE INDICATED FACTOR
     '                  ')
   54 CALL PSNAP
      CALL PWAIT(0.0.1.0)
      CALL GREPLX(TEXT.NS)
      INDEX=0
      CALL INX(TEXT.INDEX.NS.DUMMY.&900)
      VARMIN(KK)=DUMMY
      CALL INX(TEXT.INDEX.NS.DUMMY.&900)
      VARMAX(KK)=DUMMY
      IF(VARMIN(KK).LT.VARMAX(KK)) GO TO 56
      TEMP=VARMIN(KK)
      VARMIN(KK)=VARMAX(KK)
      VARMAX(KK)=TEMP
   56 IF(VARMIN(KK).GE.CONCOM(IVAR(KK).1).AND.VARMAX(KK).LE.
     1 CONCOM(IVAR(KK).LVL(IVAR(KK)))) GO TO 57
      CALL PSTUFF(14.'VALUES OUTSIDE OF THE RANGE HAVE BEEN CHOSEN.
     1                  ')
      CALL PSTUFF(15.'DEPRESS KEY 1 TO CONTINUE OR KEY 2 TO RESELECT THE
     1 VALUES           ')
      CALL PSNAP
      CALL PWAIT(1.0.0.0)
      CALL PEWLNE(14)
      CALL PEWLNE(15)
      IF(KEY.EQ.2) GO TO 54
   57 CALL PPLUCK(ILN.BUS)
      DO 156 IPUT=1.16
      BUS(IPUT+3)=TEXT(IPUT)
  156 CONTINUE
      CALL PSTUFF(ILN.BUS)
  155 CONTINUE
      IF(NF.EQ.2) GO TO 400
      DO 60 I=1.NF
      J=2*I-1
      IF(I.EQ.IVAR(1).OR.I.EQ.IVAR(2)) GO TO 60
      WRITE(NDM.1000) NAME(J).NAME(J+1)
```

```
      CALL FETCH(TEXT.MS)
      CALL PSTUFF(14.TEXT)
   59 CALL PSNAP
      CALL PWAIT(0.0.1.0)
      CALL OREPLX(BUS.NS)
      CALL PPLUCK(I+3.TEXT)
      DO 157 K=1.16
      TEXT(K+3)=BUS(K)
  157 CONTINUE
      CALL PSTUFF(I+3.TEXT)
      INDEX=0
      CALL INX(BUS.INDEX.NS.DUMMY.&960)
      RAWVAR(I)=DUMMY
      IF(RAWVAR(I).GE.CONCOM(I.1).AND.RAWVAR(I).LE.CONCOM(I.LVL(I)))
     1 GO TO 60
      CALL PSTUFF(15.'THE VALUE IS OUTSIDE OF THE RANGE.
     1              ')
      CALL PSTUFF(16.'DEPRESS KEY 1 TO CONTINUE OR KEY 2 TO RESELECT THE
     1 VALUE.       ')
      CALL PSNAP
      CALL PWAIT(1.0.0.0)
      CALL PEWLNE(15)
      CALL PEWLNE(16)
      IF(KEY.EQ.2) GO TO 59
   60 CONTINUE
  400 CALL PSTUFF(14.'ENTER THE PREDICTED RESPONSE FOR WHICH YOU WISH TO
     1 PLOT A CONTOUR   ')
      CALL PSNAP
      CALL PWAIT(0.0.1.0)
      CALL OREPLX(TEXT.NS)
      INDEX=0
      CALL INX(TEXT.INDEX.NS.DUMMY.&900)
      RESP=DUMMY
      CALL PSTUFF(14.'DEPRESS KEY 1 TO PLOT
     1              ')
      CALL PSNAP
      CALL PWAIT(1.0.0.0)
  913 CALL REFILL
      KNTPLO=1
      TEXT(1)=PLOTNO(1)
      TEXT(2)=PLOTNO(2)
      TEXT(3)=BLANK
      TEXT(4)=RESPAL
      TEXT(5)=ONSE
      IF(NF.EQ.2) GO TO 191
      KKK=6
      DO 190 JJ=1.NF
      IF(JJ.EQ.IVAR(1).OR.JJ.EQ.IVAR(2)) GO TO 190
      KK=2*JJ-1
      TEXT(KKK)=BLANK
      TEXT(KKK+1) =NAME(KK)
      TEXT(KKK+2)=NAME(KK+1)
      KKK=KKK+3
  190 CONTINUE
```

```
191 NCHAR=12*NF-4
    TOP=4055.
    CALL ORCHAR('BP',TEXT,NCHAR,23.,TOP,IER)
    NCHAR=NCHAR+2
    RESRNG=VARMAX(2)-VARMIN(2)
    RNG=VARMAX(1)-VARMIN(1)
    DEL=RNG/50.
    NX=165
    CALL UVOO(NX)
    MF=NF-1
    DO 409 III=1,4
    NORDR(III)=III
409 CONTINUE
    K=1
    DO 410 I=1,NF
    IF(I.EQ.IVAR(1).OR.I.EQ.IVAR(2)) GO TO 410
    NORDR(K)=I
    K=K+1
410 CONTINUE
411 NORDR(MF)=IVAR(1)
    NORDR(NF)=IVAR(2)
    DO 420 I=1,4
    LVT(I)=LVL(NORDR(I))
420 CONTINUE
    DO 430 J=1,NFG
    CALL ANAL(J,IX(1),IX(2),IX(3),IX(4))
    DO 440 K=1,4
    JX(K)=IX(NORDR(K))
440 CONTINUE
    NY=LTNEW(LVT,JX(1),JX(2),JX(3),JX(4))
    C(NY)=A(J)
430 CONTINUE
    IF(NF.EQ.2) GO TO 201
    MFF=NF-2
    DO 200 J=1,MFF
    JT=NORDR(J)
    JTLVL=LVL(JT)-1
    VAL(J)=POLY(COEF(1,JT),RAWVAR(JT),JTLVL)
200 CONTINUE
    NCOF=NFG
    DO 460 K=1,MFF
    LSTEP=LVT(K)
    ICT=1
    DO 470 J=1,NCOF,LSTEP
    C(ICT)=BPOLY(C(J),VAL(K),LSTEP)
    ICT=ICT+1
470 CONTINUE
    NCOF=NCOF/LSTEP
460 CONTINUE
201 C(1)=C(1)-RESP
    XVAR=VARMIN(1)
    NCOF=LVT(MF)*LVT(NF)
    LSTEP=LVT(MF)
    KPT=0
```

```
      DO 220 J=1,51
      XVAL=POLY(COEF(1,IVAR(1)),XVAR,LVT(MF)-1)
      ICT=1
      DO 480 JJ=1,NCOF,LSTEP
      B(ICT)=BPOLY(C(JJ),XVAL,LSTEP)
      ICT=ICT+1
  480 CONTINUE
      CALL ROOTER(B,ROOT,NUMRT,LVT(NF))
      IF(NUMRT.EQ.0) GO TO 218
      DO 490 KJJ=1,NUMRT
      YVAR=BPOLY(CINV,ROOT(KJJ),LVT(NF))
      IF(YVAR.LT.VARMIN(2)) GO TO 490
      IF(YVAR.GT.VARMAX(2)) GO TO 490
      CALL PUTUV(XVAR,YVAR,KERROR)
      KPT=KPT+1
  490 CONTINUE
  218 XVAR=XVAR+DEL
  220 CONTINUE
      IF(KNTPLO.NE.1) GO TO 700
      WRITE(NOM,1002) VARMIN(1)
      CALL FETCH(BUS,MC)
      CALL GRCHAR('BP',BUS,MC,364.,460.,IER)
      WRITE(NOM,1002) VARMAX(1)
      CALL FETCH(BUS,MC)
      CALL GRCHAR('BP',BUS,MC,3364.,460.,IER)
      VARINC(1)=RNG/10.
      WRITE(NOM,1003) VARINC(1)
      CALL FETCH(BUS,MC)
      CALL GRCHAR('BP',BUS,MC,1640.,440.,IER)
      BUS(1)=NAME(2*IVAR(1)-1)
      BUS(2)=NAME(2*IVAR(1))
      CALL GRCHAR('BP',BUS,8,1976.,520.,IER)
      VARINC(2)=(VARMAX(2)-VARMIN(2))/10.
      WRITE(NOM,1002) VARMIN(2)
      CALL FETCH(BUS,MC)
      CALL GRCHAR('BP',BUS,MC,0.,560.,IER)
      WRITE(NOM,1002) VARMAX(2)
      CALL FETCH(BUS,MC)
      CALL GRCHAR('BP',BUS,MC,0.,3560.,IER)
      WRITE(NOM,1004)
      CALL FETCH(BUS,MC)
      CALL GRCHAR('BP',BUS,MC,0.,2140.,IER)
      WRITE(NOM,1002) VARINC(2)
      CALL FETCH(BUS,MC)
      CALL GRCHAR('BP',BUS,MC,0.,2060.,IER)
      BUS(1)=NAME(2*IVAR(2)-1)
      BUS(2)=NAME(2*IVAR(2))
      CALL GRCHAR('BP',BUS,8,0.,2220.,IER)
      CALL GRCHAR('BP','DEPRESS KEY 1 TO RETURN TO OPTION TABLE',
     1 39,500.,340.,IER)
      CALL GRCHAR('BP','OR ENTER NEW RESPONSE (NEW FRAME CREATED AFTER 4
     1 PLOTS.)',56,500.,250.,IER)
  700 DO 699 IBL=1,20
  699 TEXT(IBL)=BLANK
```

```
      TEXT(1)=ALIST(KNTPLO)
      TEXT(2)=BLANK
      WRITE(NDM,1001) RESP
      CALL FETCH(BUS,MC)
      TEXT(3)=BUS(1)
      TEXT(4)=BUS(2)
      TEXT(5)=BUS(3)
      KK=6
      IF(NF.EQ.2) GO TO 720
      DO 705 J=1,NF
      IF(J.EQ.IVAR(1).OR.J.EQ.IVAR(2)) GO TO 705
      WRITE(NDM,1001) RAWVAR(J)
      CALL FETCH(BUS,MC)
      TEXT(KK)=BUS(1)
      TEXT(KK+1)=BUS(2)
      TEXT(KK+2)=BUS(3)
      KK=KK+3
  705 CONTINUE
  720 IF(KPT.NE.0) GO TO 725
      TEXT(KK)=BAD(1)
      TEXT(KK+1)=BAD(2)
      TEXT(KK+2)=BAD(3)
      KK=KK+3
  725 TOP=TOP-80.
      NCHAR=4*KK
      CALL GRCHAR('BP',TEXT,NCHAR,23.,TOP,IER)
      CALL GRAXES(VARMIN(1),VARMAX(1),VARMIN(2),VARMAX(2),700,3700,
     1600,3600)
      IF(KNTPLO.NE.1) GO TO 760
      CALL GRGRID(RNG,RESRNG,'A',IFLAG)
      XLBL=VARMIN(1)
      YLBL=VARMIN(2)
      DO 512 LBL=1,9
      XLBL=XLBL+VARINC(1)
      YLBL=YLBL+VARINC(2)
      CALL GRCHAR('BP','\',1,XLBL,VARMIN(2),IER)
      CALL GRCHAR('BP','-',1,VARMIN(1),YLBL,IER)
  512 CONTINUE
  760 IF(KPT.EQ.0) GO TO 770
      CALL GRPLOT('BP','B','A',1,KPT,PLIST(KNTPLO),'O',IER)
  770 CALL PWAIT(1.0,1,0)
      CALL UV99
      IF(ITYPE.EQ.1) GO TO 780
      KNTPLO=KNTPLO+1
      CALL GREPLX(TEXT,NS)
      C(1)=C(1)+RESP
      INDEX=0
      CALL INX(TEXT,INDEX,NS,DUMMY,4900)
      RESP=DUMMY
      CALL GRAXES(0.,4095.,80.,4095.,0,4095,80,4095)
      IF(KNTPLO.GT.4) GO TO 913
      CALL UV00(NX)
      GO TO 201
  780 CALL REFILL
```

```
      CALL RLSEP
      RETURN
  900 CONTINUE
  960 CONTINUE
      WRITE(6,2001)
 2001 FORMAT(20X,'BAD CONVERSION')
      CALL RLSEP
      RETURN
      END
```

```
SUBROUTINE ANAL (NN.IA.IB.IC.ID)
COMMON /YATDAT/ DUM1.DUM2.DUM3.DUM4.LVL(4)
N = NN
NFG = LVL(1)*LVL(2)*LVL(3)
ID = (N-1)/NFG + 1
N = MOD(N-1.NFG) + 1
NFG = LVL(1)*LVL(2)
IC = (N-1)/NFG + 1
N = MOD(N-1.NFG) + 1
IB = (N-1)/LVL(1) + 1
IA = MOD(N-1.LVL(1) ) + 1
RETURN
END
```

```
SUBROUTINE BNAL(NN.IA.IB.IC.ID.LVL)
DIMENSION LVL(1)
N = NN
NFG=LVL(1)*LVL(2)*LVL(3)
ID = (N-1)/NFG + 1
N = MOD(N-1.NFG) + 1
NFG = LVL(1)*LVL(2)
IC = (N-1)/NFG + 1
N = MOD(N-1.NFG) + 1
IB = (N-1)/LVL(1) + 1
IA = MOD(N-1.LVL(1) ) + 1
RETURN
END
```

```
FUNCTION LT(I1,I2,I3,I4)
COMMON /YATDAT/ DUM1,DUM2,DUM3,DUM4,LVL(4)
LT=(((I4-1)*LVL(3)+I3-1)*LVL(2)+I2-1)*LVL(1)+I1
RETURN
END
```

```
FUNCTION LTNEW(LVL,I1,I2,I3,I4)
DIMENSION LVL(1)
LT=(((I4-1)*LVL(3)+I3-1)*LVL(2)+I2-1)*LVL(1)+I1
LTNEW=LT
RETURN
END
```

```
      FUNCTION BPOLY(A.X.N)
      DIMENSION A(1)
      BPOLY=0.
      IF(N.LT.1) RETURN
      BPOLY=A(N)
      IF(N.EQ.1) RETURN
      NDEG=N-1
      DO 1 J=1.NDEG
      JM=N-J
    1 BPOLY=X*BPOLY+A(JM)
      RETURN
      END
```

```
      FUNCTION POLY(A,X,N)
      DIMENSION A(1)
      POLY=0.
      IF(N.LT.0) RETURN
      POLY=A(1)
      IF(N.EQ.0) RETURN
      DO 1 J=1,N
    1 POLY=X*POLY+A(J+1)
      RETURN
      END
```

```
      SUBROUTINE ROOTER(S,R,NR,LVL)
      DIMENSION S(1),R(1),B(4),A(4)
      DO 5 I=1,LVL
      A(I)=S(I)
    5 CONTINUE
      NR=0
      IND=LVL-1
      GO TO (10,20,30),IND
   10 IF(A(2).EQ.0.) RETURN
      NR=NR+1
      R(NR)=-A(1)/A(2)
      RETURN
   20 IF(A(3).EQ.0.) GO TO 10
      TERM=A(2)*A(2)-4.0*A(3)*A(1)
      IF(TERM) 22,24,26
   24 NR=NR+1
      R(NR)=-A(2)/(2.0*A(3))
   22 RETURN
   26 NR=NR+1
      R(NR)=(-A(2)-SQRT(TERM))/(2.0*A(3))
      NR=NR+1
      R(NR)=(-A(2)+SQRT(TERM))/(2.0*A(3))
      RETURN
   30 IF(A(4).EQ.0.) GO TO 20
      X=0.
      DO 40 I=1,100
      XSAV=X
      B(3)=A(4)
      B(2)=A(3)+X*B(3)
      B(1)=A(2)+X*B(2)
      DF=A(1)+X*B(1)
      TEMP=B(2)+X*B(3)
      TEMP=B(1)+X*TEMP
      IF(TEMP.NE.0.) GO TO 35
      X=X+.5
      GO TO 40
   35 X=X-DF/TEMP
      IF(ABS(X-XSAV).LE.1.E-5) GO TO 50
   40 CONTINUE
      NR=0
      RETURN
   50 NR=1
      R(1)=X
      A(3)=B(3)
      A(2)=B(2)
      A(1)=B(1)
      GO TO 20
      END
```

```
      SUBROUTINE SPOLY(B.VAL.ISTOP.NORDR.A)
      DOUBLE PRECISION DSSQ.DER
      DIMENSION A(1).VAL(1).NORDR(1).B(1)
      COMMON /YATDAT/ DSSQ.DER.
     3LVL(4).NFG.IRAW.DAT(256).NEND(4.4).TTL(20).
     1VNAM(2).NAME( 8).FMT(80).ICODE.IL(6).         NIN.NOUT.NOX.NF.
     2NVBL.NREP.CONCOM(4.4).NR(256).FA.FB.NO1(256).IV.NW
      IF(ISTOP.GT.NF) ISTOP=NF
      DO 10 JK=1.NFG
   10 A(JK)=B(JK)
      NCOF=NFG
      DO 50 K=1.ISTOP
      JT=NORDR(K)
      LSTEP=LVL(JT)
      ICT=1
      DO 40 J=1.NCOF.LSTEP
      A(ICT)=BPOLY(A(J).VAL(K).LSTEP)
      ICT=ICT+1
   40 CONTINUE
      NCOF=NCOF/LSTEP
   50 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE XCHNGE(JF,COF)
      DIMENSION      COF(1),LVT(6),IX(6)
      DOUBLE PRECISION DSSQ,DER
      COMMON /YATDAT/ DSSQ,DER,
     3LVL(4),NFG,IRAW,DAT(256),NEND(4,4),TTL(20),
     1VNAM(2),NAME( 8),FMT(80),ICODE,IL(6),        NIN,NOUT,NDX,NF,
     2NVBL,NREP,CONCOM(4,4),NR(256),FA,FB,ND1(256),IV,NW,
     4COEF(4,6),A(512)
      DO 5 K=1,NF
    5 LVT(K)=LVL(K)
      LVT(NF)=LVL(JF)
      LVT(JF)=LVL(NF)
      DO 10 J=1,NFG
      CALL ANAL(J,IX(1),IX(2),IX(3),IX(4))
      ITEMP=IX(NF)
      IX(NF)=IX(JF)
      IX(JF)=ITEMP
      NY=LTNEW(LVT,IX(1),IX(2),IX(3),IX(4))
      COF(NY)=A(J)
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ZTORAW(C.J)
      COMMON /YATDAT/ DUMA(4).LVL(4).DUMB(397).X(4.4)
      DIMENSION C(1)
      KGO=LVL(J)-1
      GO TO (10.20.30).KGO
   10 C(2)=(X(J.2)-X(J.1))/2.0
      C(1)=(X(J.2)+X(J.1))/2.0
      RETURN
   20 C(3)=(X(J.3)-2.0*X(J.2)+X(J.1))/2.0
      C(2)=(X(J.3)-X(J.1))/2.0
      C(1)=X(J.2)
      RETURN
   30 C(4)=(9.0*X(J.4)-27.0*X(J.3)+27.0*X(J.2)-9.0*X(J.1))/16.
      C(3)=9.0*(X(J.4)-X(J.3)-X(J.2)+X(J.1))/16.0
      C(2)=(-X(J.4)+27.0*X(J.3)-27.0*X(J.2)+X(J.1))/16.0
      C(1)=(-X(J.4)+9.0*X(J.3)+9.0*X(J.2)-X(J.1))/16.0
      RETURN
      END
```

```
      SUBROUTINE DSMULT(A.N.C.B)
      DIMENSION A(1).B(1)
C OR  DOUBLE PRECISION A(1).B(1).C
      IF(N.LT.0 .OR. N.GE.120 )  GO TO 95
      NP = N+1
      DO 1 I=1.NP
 1    B(I) = A(I)
      B(NP+1) =-C*B(NP)
      IF(N.EQ.0) GO TO 99
      DO 2 J=1.N
      JX = N-J+1
 2    B(JX+1) = B(JX+1)-C*B(JX)
      GO TO 99
 95   WRITE(6.91) N
 91   FORMAT(1H0.5X.31HDSMULT DOES NOT WORK FOR DEGRE  .I5)
 99   RETURN
      END
```

```
      SUBROUTINE DWMULT(A,N,C,B)
C  TAKE AN EQUATION A(1)*(X-C(1))*(X-C(2))*....(X-C(N)) + A(2)*(X-C(1))
C   *(X-C(2))*...*(X-C(N-1)) + ... + A(N)*(X-C(1)) + A(N+1). CONVERT INT
C   B(1)*X**N + B(2)*X**(N-1) +...+B(N+1)
      DIMENSION A(1),C(1),B(1)
C OR  DOUBLE PRECISION A(1),C(1),B(1),HOLD
      IF(N.LT.0 .OR. N.GE.120) GO TO 95
      NP = N+1
      DO 1 I=1,NP
 1    B(I) = A(I)
      IF (N.EQ.0) GO TO 99
      DO 2 J=1,N
      HOLD=B(J+1)
      JX = N-J+1
      CALL DGMULT(B,J-1,C(JX),B)
 2    B(J+1) = B(J+1) + HOLD
      GO TO 99
 95   WRITE(6,91) N
 91   FORMAT(1H0,10X,31HDWMULT DOES NOT WORK FOR DEGREE ,I5)
 99   RETURN
      END
```

```
      FUNCTION ORTH(Z.I.N)
      M=N-1
      GO TO (10.20.30).M
   10 GO TO (12.14).I
   12 ORTH=1.0
      RETURN
   14 ORTH=Z
      RETURN
   20 GO TO (12.14.22).I
   22 ORTH=3.0*Z*Z-2.0
      RETURN
   30 GO TO (12.32.34.36).I
   32 ORTH=3.0*Z
      RETURN
   34 ORTH=(9.0*Z*Z-5.0)/4.0
      RETURN
   36 ORTH=((45.0*Z*Z-41.0)*Z)/4.0
      RETURN
      END
```

```
      EXTERNAL EOBINT.PFINT
      DOUBLE PRECISION P.    DUMMY.PVAL(4).XMIN.XMAX.XINC.YMIN.YMAX
      DOUBLE PRECISION FMT.FMT1.FMT2.FORM(7).OP.CLO.FORM1.FORM2
      REAL NAME(2)
      INTEGER*2 INFUN(34).INSERT
      DIMENSION X(15).Y(15).C(4.11.4).ICH(4).XCOL(11).ANS(11.4)
      DIMENSION YSAV(15.4).RMT(1)
      DIMENSION TEXT(20).FUN(17)
      DIMENSION MMM(3)
      COMMON KEY.IOVLY.ITYPE.JTYPE(4).MASK
      EQUIVALENCE (FMT.RMT(1))
      EQUIVALENCE (INFUN(1).FULL)
      EQUIVALENCE (NOM.NT)
      DATA BLANK/' '/
      DATA INSERT /'Y='/
      DATA OP.CLO.FORM1.FORM2 /'('.'1X)'.'F12.5.'.'E12.5.'/
      DATA FMT1.FMT2.ICH /'(F9.2)'.'(E9.3)'.'1'.'2'.'3'.'4'/
      DATA MMM /Z60000001.Z7E000009.Z70000001/
      REWIND 8
      CALL PLOTS(IDUM.JDUM.8)
      M=11
      NC=68
      NOM=4
      NPARM=0
      IFLAG=0
      NZ=0
      CALL GRINIT('8')
      CALL GCEOB(EOBINT)
      CALL NEWPGE
      MASK=MMM(1)
      CALL GCPFK(MASK.PFINT)
      CALL STUFF(2.'THIS UNIT WILL PLOT ANY EXPRESSION CURRENTLY AVAILAB
     1LE            ')
      CALL STUFF(3.'IN THE CALCULATOR MODE.
     1              ')
      CALL STUFF(5.'THE FUNCTION CAN BE OF THE FORM
     1              ')
      CALL STUFF(7.'      Y = F(X) .  OR
     1              ')
      CALL STUFF(8.'      Y = F(Z)  WHERE  Z = G(X)
     1              ')
      CALL STUFF(10.'IN ADDITION. THE EXPRESSION MAY BE PLOTTED FOR UP
     1              ')
      CALL STUFF(11.'TO FOUR VALUES OF A PARAMETER. SAY P.
     1              ')
      CALL STUFF(13.'DEPRESS KEY 31 TO TERMINATE THE SESSION.
     1              ')
      CALL STUFF(15.'IF YOU WISH TO SPECIFY A VARIABLE PARAMETER
     1              ')
      CALL STUFF(16.'DEPRESS KEY 2; OTHERWISE. DEPRESS KEY 1 TO CONTINUE
     1              ')
      CALL SNAP
      CALL TWAIT(1.0.0.0)
      IF(KEY.EQ.1) GO TO 10
```

BLOWUP - MAIN PROGRAM

```
      LN=18
   61 CALL STUFF(LN,'ENTER THE NAME BY WHICH YOU WILL
     1                ')
      LN=LN+1
      CALL STUFF(LN,'REFER TO THE PARAMETER
     1                ')
      CALL SNAP
      CALL TWAIT(0.0,1.0)
      CALL ORRPLX(TEXT,NC)
      NAME(1)=TEXT(1)
      NAME(2)=TEXT(2)
      LN=LN+2
      CALL STUFF(LN,'ENTER, SEPARATED BY COMMAS, UP TO FOUR VALUES FOR
     1                ')
      LN=LN+1
      CALL STUFF(LN,TEXT)
      LN=LN+1
      CALL SNAP
      NBACK=1
   50 CALL TWAIT(0.0,1.0)
      CALL ORRPLX(TEXT,NC)
      CALL STUFF(LN,TEXT)
      LN=LN+2
      CALL SNAP
      INDEX=0
      NPARM=0
      DO 5 I=1,4
      IF(INDEX.LT.0) GO TO 16
      CALL INX(TEXT,INDEX,NC,DUMMY,&1000)
      PVAL(I)=DUMMY
      NPARM=NPARM+1
    5 CONTINUE
   16 IF(IFLAG.EQ.1) GO TO 100
      GO TO 15
   10 LN=18
   15 CALL STUFF(LN,'IF YOU WISH TO SPECIFY Z AS A FUNCTION OF X
     1                ')
      LN=LN+1
      CALL STUFF(LN,'DEPRESS KEY 2; OTHERWISE, DEPRESS KEY 1 TO CONTINUE
     1                ')
      LN=LN+2
      CALL SNAP
      CALL TWAIT(1.0,0.0)
      IF(KEY.EQ.1) GO TO 31
   21 CALL STUFF(LN,'ENTER THE FUNCTION IN THE FORM  Z = G(X)
     1                ')
      LN=LN+1
      NZ=1
      CALL SNAP
      CALL TWAIT(0.0,1.0)
      CALL ORRPLX(INFUN,NC)
      IF(IFLAG.EQ.1) GO TO 100
      CALL STUFF(LN,INFUN)
      LN=LN+2
```

```
31 CALL STUFF(LN,'ENTER, SEPARATED BY COMMAS, THE MINIMUM AND MAXIMUM
  1 VALUES FOR X     ')
   LN=LN+1
   CALL SNAP
   NBACK=2
60 CALL TWAIT(0,0,1,0)
   CALL GRRPLX(TEXT,NC)
   CALL STUFF(LN,TEXT)
   LN=LN+1
   INDEX=0
   CALL INX(TEXT,INDEX,NC,XMIN,&1000)
   CALL INX(TEXT,INDEX,NC,XMAX,&1000)
   IF(IFLAG.EQ.1) GO TO 100
41 CALL STUFF(LN,'ENTER THE INCREMENT FOR X.
  1                   ')
   LN=LN+1
   NBACK=3
   CALL SNAP
70 CALL TWAIT(0,0,1,0)
   CALL GRRPLX(TEXT,NC)
   CALL STUFF(LN,TEXT)
   LN=LN+2
   INDEX=0
   CALL INX(TEXT,INDEX,NC,XINC,&1000)
   IF(IFLAG.EQ.1) GO TO 100
51 CALL STUFF(LN,'ENTER, SEPARATED BY COMMAS, THE MINIMUM AND MAXIMUM
  1 VALUES FOR Y     ')
   LN=LN+1
   CALL SNAP
   NBACK=4
80 CALL TWAIT(0,0,1,0)
   CALL GRRPLX(TEXT,NC)
   CALL STUFF(LN,TEXT)
   LN=LN+2
   INDEX=0
   CALL INX(TEXT,INDEX,NC,YMIN,&1000)
   CALL INX(TEXT,INDEX,NC,YMAX,&1000)
   IF(IFLAG.EQ.1) GO TO 100
11 CALL STUFF(LN,'ENTER THE FUNCTION IN THE FORM  Y = F(X)
  1                   ')
   LN=LN+1
   CALL STUFF(LN,'OR  Y = F(Z)  IF Z IS DEFINED
  1                   ')
   LN=LN+1
   CALL SNAP
   CALL TWAIT(0,0,1,0)
   CALL GRRPLX(FUN,NC)
   IF(IFLAG.EQ.1) GO TO 100
   CALL STUFF(LN,FUN)
   LN=LN+2
   CALL STUFF(LN,'DEPRESS KEY 1 TO CONTINUE.
  1                   ')
   CALL SNAP
   CALL TWAIT(1,0,0,0)
```

```
      GO TO 100
 1000 CALL STUFF(LN.'AN INCORRECT NUMBER HAS BEEN ENTERED.   TRY AGAIN.
     1                  ')
      CALL SNAP
      GO TO (50.60.70.80). NBACK
  100 CALL NEWPGE
      CALL GCPFK(MASK.PFINT)
      MASK=MMM(2)
      CALL STUFF( 1.'           SUMMARY
     1                  ')
      WRITE(NOM.500) FUN
  500 FORMAT(' 1. '.17A4)
      CALL FETCH(TEXT.MC)
      LN=3
      CALL STUFF(LN.TEXT)
      LN=LN+1
      IF(NZ.EQ.0) GO TO 110
      WRITE(NOM.510) INFUN
  510 FORMAT(' 2. '.34A2)
      CALL FETCH(TEXT.MC)
      CALL STUFF(LN.TEXT)
      LN=LN+1
      GO TO 120
  110 CALL STUFF(LN.' 2. NO Z = G(X) SPECIFIED
     1                  ')
      LN=LN+1
  120 WRITE(NOM.520) XMIN.XMAX
  520 FORMAT(' 3. MIN FOR X = '.E14.7.' MAX FOR X = '.E14.7.11X)
      CALL FETCH(TEXT.MC)
      CALL STUFF(LN.TEXT)
      LN=LN+1
      WRITE(NOM.530) XINC
  530 FORMAT(' 4. INCREMENT FOR X = '.E14.7.32X)
      CALL FETCH(TEXT.MC)
      CALL STUFF(LN.TEXT)
      LN=LN+1
      WRITE(NOM.540) YMIN.YMAX
  540 FORMAT(' 5. MIN FOR Y = '.E14.7.' MAX FOR Y = '.E14.7.11X)
      CALL FETCH(TEXT.MC)
      CALL STUFF(LN.TEXT)
      LN=LN+1
      IF(NPARM.EQ.0) GO TO 130
      WRITE(NOM.550) NAME
  550 FORMAT(' 6. THE VALUES FOR '.2A4.' ARE:'.36X)
      CALL FETCH(TEXT.MC)
      CALL STUFF(LN.TEXT)
      LN=LN+1
      DO 125 K=1.17
      TEXT(K)=BLANK
  125 CONTINUE
      WRITE(NOM.560) (PVAL(J).J=1.NPARM)
  560 FORMAT(4X.4E16.7)
      CALL FETCH(TEXT.MC)
      CALL STUFF(LN.TEXT)
```

```
      LN=LN+2
      GO TO 140
  130 CALL STUFF(LN,' 6. NO PARAMETER SPECIFIED.
     1                 ')
      LN=LN+2
  140 CALL STUFF(LN,'DEPRESS THE KEY CORRESPONDING TO ANY ITEM YOU WISH
     1TO CHANGE        ')
      LN=LN+2
      CALL STUFF(LN,'OR DEPRESS KEY 28 TO PLOT THE FUNCTION
     1                 ')
      CALL SNAP
      CALL TWAIT(1,0,0,0)
      IF(KEY.EQ.28) GO TO 607
      IFLAG=1
      GO TO (11,21,31,41,51,61), KEY
  607 CALL GRRLSE
  600 CONTINUE
      CALL INITP
      MASK=MMM(3)
      CALL PFKP(MASK,PFINT)
      CALL EOBP(EOBINT)
      CALL GRAXES(0.,4092.,100.,4092.,0,4092,100,4092)
      HT=4050.
      CALL GRCHAR('BP',FUN,NC,800.,HT,K)
      IF(NZ.EQ.0) GO TO 610
      HT=HT-100.
      CALL GRCHAR('BP',INFUN,NC,800.,HT,K)
  610 IF(NPARM.EQ.0) GO TO 620
      HT=HT-100.
      CALL GRCHAR('BP','  PLOT NO.     PARAMETER',24,800.,HT,K)
      IND=1
      DO 612 J=1,NPARM
      IF(PVAL(J).LT..001.OR.PVAL(J).GT.9999.999) IND=2
  612 CONTINUE
      DO 614 K=1,NPARM
      HT=HT-100.
      GO TO (616,618) , IND
  616 WRITE(NDM,900) K,PVAL(K)
      GO TO 619
  618 WRITE(NDM,910) K,PVAL(K)
  900 FORMAT(I6,8X,F10.4,44X)
  910 FORMAT(I6,7X,F14.7,41X)
  619 CALL FETCH(TEXT,MC)
      CALL GRCHAR('BP',TEXT,MC,800.,HT,KFLAG)
  614 CONTINUE
  620 AMIN=XMIN
      AMAX=XMAX
      BMIN=YMIN
      BMAX=YMAX
      AINC=(AMAX-AMIN)/10.
      BINC=(BMAX-BMIN)/10.
      CALL GRCHAR('BP','PFK: 1 = SUMMARY, 2 = NEW Y-AXIS, 3 = BLOW-UP, 3
     11 = STOP',56,50.,200.,K)
      FMT=FMT1
```

```
      IF(ABS(AMIN).LT..01.OR.ABS(AMIN).GT.999.999) FMT=FMT2
      WRITE(NT,RMT) AMIN
      CALL FETCH(TEXT,MC)
      CALL GRCHAR('BP',TEXT,MC,575.,400.,K)
      FMT=FMT1
      IF(ABS(AMAX).LT..01.OR.ABS(AMAX).GT.999.999) FMT=FMT2
      WRITE(NT,RMT) AMAX
      CALL FETCH(TEXT,MC)
      CALL GRCHAR('BP',TEXT,MC,3475.,400.,K)
      FMT=FMT1
      IF(ABS(BMIN).LT..01.OR.ABS(BMIN).GT.999.999) FMT=FMT2
      WRITE(NT,RMT) BMIN
      CALL FETCH(TEXT,MC)
      CALL GRCHAR('BP',TEXT,MC,175.,500.,K)
      FMT=FMT1
      IF(ABS(BMAX).LT..01.OR.ABS(BMAX).GT.999.999) FMT=FMT2
      WRITE(NT,RMT) BMAX
      CALL FETCH(TEXT,MC)
      CALL GRCHAR('BP',TEXT,MC,175.,3420.,K)
      CALL GRCHAR('BP','INC = ',6,1900.,400.,K)
      CALL GRCHAR('BP','INC = ',6,175.,2000.,K)
      FMT=FMT1
      IF(ABS(AINC).LT..01.OR.ABS(AINC).GT.999.999) FMT=FMT2
      WRITE(NT,RMT) AINC
      CALL FETCH(TEXT,MC)
      CALL GRCHAR('BP',TEXT,MC,2250.,400.,K)
      FMT=FMT1
      IF(ABS(BINC).LT..01.OR.ABS(BINC).GT.999.999) FMT=FMT2
      WRITE(NT,RMT) BINC
      CALL FETCH(TEXT,MC)
      CALL GRCHAR('BP',TEXT,MC,175.,1900.,K)
      CALL GRCHAR('BP','X',1,2300.,300.,K)
      CALL GRAXES(XMIN,XMAX,YMIN,YMAX,800,3800,500,3500)
      DELX=(XMAX-XMIN)/2.0
      DELY=(YMAX-YMIN)/2.0
      CALL GRGRID(DELX,DELY,'A',K)
      NUMPLO=NPARM
      IF(NUMPLO.EQ.0) NUMPLO=1
      DO 700 J=1,NUMPLO
      CALL UVOO(15)
      IF(NPARM.EQ.0) GO TO 333
      CALL ENSYM(NAME,PVAL(J),KK)
  333 XDUM=XMIN
      MK=0
      DO 710 I=1,11
      CALL ENSYM('X        ',XDUM,KK)
      IF(NZ.EQ.0) GO TO 720
      IHOLD=INFUN(1)
      INFUN(1)=INSERT
      ZVAL=EVAL(INFUN,68,KODET,MERR)
      INFUN(1)=IHOLD
      CALL ENSYM('Z        ',ZVAL,KK)
  720 YDUM=EVAL(FUN,68,KODET,MERR)
      IF(YDUM.LT.YMIN.OR.YDUM.GT.YMAX) GO TO 1752
```

```
      CALL PUTUV(XDUM,YDUM,KKK)
      MK=MK+1
      WRITE(6,1717) XDUM,YDUM
 1717 FORMAT(2E20.7)
 1752 X(I)=XDUM
      Y(I)=YDUM
      XDUM=XDUM+AINC
  710 CONTINUE
      DO 730 I=1,11
      YSAV(I,J)=Y(I)
  730 CONTINUE
      CALL SPLCON(X,Y,M,C(1,1,J))
      CALL GRPLOT('BP','B','B',1,MK,ICH(J),'O',KKK)
      CALL UV99
      CALL UV00(210)
      NPTS=0
      XOUT=XMIN
      IF((XMAX-XMIN)/XINC.GT.200.) XINC=(XMAX-XMIN)/200.
  750 CALL SPLINE(X,YSAV(1,J),M,C(1,1,J),XOUT,YOUT)
      XDUM=XOUT
      YDUM=YOUT
      IF(YDUM.LT.YMIN.OR.YDUM.GT.YMAX) GO TO 752
      CALL PUTUV(XDUM,YDUM,KKK)
  752 XOUT=XOUT+XINC
      NPTS=NPTS+1
      IF(XOUT.LT.XMAX) GO TO 750
      CALL GRPLOT('BP','A','B',1,NPTS,'*','O',KKK)
      CALL UV99
  700 CONTINUE
      CALL PWAIT(1,0,0,0)
      CALL GRAXES(0.,4092.,100.,4092.,0,4092,100,4092)
      GO TO (915,920,930),KEY
  915 CALL REFILL
      CALL RLSEP
      CALL GRINIT('B')
      CALL OCEOB(EOBINT)
      GO TO 100
  920 CALL GRCHAR('BP','ENTER NEW MINIMUM, MAXIMUM',26,50.,100.,K)
      CALL GREPLY(TEXT,NC)
      CALL XBLANK(TEXT,NC)
      INDEX=0
      CALL INX(TEXT,INDEX,NC,DUMMY,&100)
      YMIN=DUMMY
      CALL INX(TEXT,INDEX,NC,DUMMY,&100)
      YMAX=DUMMY
      CALL REFILL
      CALL RLSEP
      GO TO 600
  930 CALL GRCHAR('BP','ENTER MINIMUM, MAXIMUM',22,50.,100.,K)
      CALL GREPLY(TEXT,NC)
      CALL REFILL
      CALL RLSEP
  933 CALL XBLANK(TEXT,NC)
      INDEX=0
```

```
      CALL INX(TEXT,INDEX,NC,DUMMY,&100)
      TMIN=DUMMY
      CALL INX(TEXT,INDEX,NC,DUMMY,&100)
      TMAX=DUMMY
      CALL GRINIT('B')
      CALL GCEOB(EOBINT)
      CALL NEWPGE
      MASK=MMM(3)
      CALL GCPFK(MASK,PFINT)
      LN=2
      CALL STUFF(LN,FUN)
      LN=LN+2
      IF(NZ.EQ.0) GO TO 1610
      CALL STUFF(LN,INFUN)
 1610 IF(NPARM.EQ.0) GO TO 1620
      CALL STUFF(LN,'  PLOT NO.      PARAMETER
     1                ')
      LN=LN+1
      DO 1614 K=1,NPARM
      GO TO (1616,1618),IND
 1616 WRITE(NDM,900) K,PVAL(K)
      GO TO 1619
 1618 WRITE(NDM,910) K,PVAL(K)
 1619 CALL FETCH(TEXT,MC)
      CALL STUFF(LN,TEXT)
      LN=LN+1
 1614 CONTINUE
 1620 CONTINUE
      LN=LN+1
      DO 1650 KB=1,17
      TEXT(KB)=BLANK
 1650 CONTINUE
      WRITE(NDM,1800) (ICH(KOUT),KOUT=1,NUMPLO)
 1800 FORMAT(2X,'X          ',4(4X,A8))
      CALL FETCH(TEXT,MC)
      CALL STUFF(LN,TEXT)
      TINC=(TMAX-TMIN)/10.
      XCOL(1)=TMIN
      DO 1905 I=1,10
      XCOL(I+1)=XCOL(I)+TINC
 1905 CONTINUE
      DO 1907 J=1,NUMPLO
      DO 1907 I=1,11
      CALL SPLINE(X,YSAV(1,J),M,C(1,1,J),XCOL(I),YOUT)
      ANS(I,J)=YOUT
 1907 CONTINUE
      FORM(1)=OP
      FORM(2)=FORM1
      DO 1910 I=1,11
      IF(ABS(XCOL(I)).LT..01.OR.ABS(XCOL(I)).GT.999.999) FORM(2)=FORM2
 1910 CONTINUE
      DO 1930 J=1,NUMPLO
      FORM(J+2)=FORM1
      DO 1920 I=1,11
```

```
      TEMP=ANS(I,J)
      IF(ABS(TEMP).LT..01.OR.ABS(TEMP).GT.999.999) FORM(J+2)=FORM2
 1920 CONTINUE
 1930 CONTINUE
      FORM(NUMPLO+3)=CLO
      DO 1950 I=1,11
      WRITE(NT,FORM) XCOL(I),(ANS(I,J),J=1,NUMPLO)
      CALL FETCH(TEXT,MC)
      CALL STUFF(LN,TEXT)
      LN=LN+1
 1950 CONTINUE
      LN=LN+1
      CALL STUFF(LN,'DEPRESS PFK 1 FOR SUMMARY
     1                   ')
      LN=LN+1
      CALL STUFF(LN,'             2 TO REPLOT
     1                   ')
      LN=LN+1
      CALL STUFF(LN,'             3 FOR FURTHER BLOW-UP
     1                   ')
      LN=LN+1
      CALL STUFF(LN,'            31 TO STOP
     1                   ')
      CALL SNAP
 4199 CALL TWAIT(1,0,0,0)
      GO TO (100,2920,2930), KEY
 2920 CALL GRRLSE
      GO TO 600
 2930 LN=LN+1
      CALL STUFF(LN,'ENTER MINIMUM, MAXIMUM
     1                   ')
      CALL SNAP
      CALL TWAIT(0,0,1,0)
      CALL GRRPLX(TEXT,MC)
      CALL GRRLSE
      GO TO 933
      END
```

```
      SUBROUTINE SPLCON(X.Y.M.C)
      DIMENSION X(1).Y(1).D(11).P(11).E(11).C(4.11)
      DIMENSION A(11.3).B(11).Z(11)
      MM=M-1
      DO 2 K=1.MM
      D(K)=X(K+1)-X(K)
      P(K)=D(K)/6.
    2 E(K)=(Y(K+1)-Y(K))/D(K)
      DO 3 K=2.MM
    3 B(K)=E(K)-E(K-1)
      A(1.2)=-1.-D(1)/D(2)
      A(1.3)=D(1)/D(2)
      A(2.3)=P(2)-P(1)*A(1.3)
      A(2.2)=2.*(P(1)+P(2))-P(1)*A(1.2)
      A(2.3)=A(2.3)/A(2.2)
      B(2)=B(2)/A(2.2)
      DO 4 K=3.MM
      A(K.2)=2.*(P(K-1)+P(K))-P(K-1)*A(K-1.3)
      B(K)=B(K)-P(K-1)*B(K-1)
      A(K.3)=P(K)/A(K.2)
    4 B(K)=B(K)/A(K.2)
      Q=D(M-2)/D(M-1)
      A(M.1)=1.+Q+A(M-2.3)
      A(M.2)=-Q-A(M.1)*A(M-1.3)
      B(M)=B(M-2)-A(M.1)*B(M-1)
      Z(M)=B(M)/A(M.2)
      MN=M-2
      DO 6 I=1.MN
      K=M-I
    6 Z(K)=B(K)-A(K.3)*Z(K+1)
      Z(1)=-A(1.2)*Z(2)-A(1.3)*Z(3)
      DO 7 K=1.MM
      Q=1./(6.*D(K))
      C(1.K)=Z(K)*Q
      C(2.K)=Z(K+1)*Q
      C(3.K)=Y(K)/D(K)-Z(K)*P(K)
    7 C(4.K)=Y(K+1)/D(K)-Z(K+1)*P(K)
      RETURN
      END
```

From Pennington [18]

```
      SUBROUTINE SPLINE(X,Y,M,C,XINT,YINT)
      DIMENSION X(1),Y(1),C(4,1)
      IF(XINT-X(1)) 7,1,2
    1 YINT=Y(1)
      RETURN
    2 K=1
    3 IF(XINT-X(K+1))6,4,5
    4 YINT=Y(K+1)
      RETURN
    5 K=K+1
      IF(M-K) 7,7,3
    6 YINT=(X(K+1)-XINT)*(C(1,K)*(X(K+1)-XINT)**2+C(3,K))
      YINT=YINT+(XINT-X(K))*(C(2,K)*(XINT-X(K))**2+C(4,K))
    7 RETURN
      END
```

From Pennington [18]

```
DOUBLE PRECISION FUNCTION POWER(DALF,DFH,DFE,DLAM)
DOUBLE PRECISION DALF,DFH,DFE,DLAM,BNC5,DX
DOUBLE PRECISION   DLAX,DFA,DFB,DN
DOUBLE PRECISION DTEMP
DN=DFH+DFE+1.D0
DLAX=0.5D0*DLAM*DN
DFA=0.5*DFH
DFB=0.5D0*DFE
DTEMP=1.D0-DALF
DX=BETAP(DTEMP,DFA,DFB)
POWER=1.D0-BNC5(DX,DFA,DFB,DLAX,1.D0)
IF(POWER.GT.999999D0) POWER=1.D0
IF(POWER.LT.1.D-6) POWER=0.D0
RETURN
END
```

```
      SUBROUTINE CHECK(*.*.MODE)
      EXTERNAL PFINT
      COMMON KEY.IOVLY.ITYPE.JTYPE(4).MASK
      DATA MALL /ZFFFFFFFF/
      IF(ITYPE.EQ.1.AND.KEY.EQ.31) GO TO 50
      IF(ITYPE.EQ.1.AND.KEY.EQ.15) GO TO 60
      IF(JTYPE(ITYPE).NE.1) GO TO 40
      IF(ITYPE.NE.1) GO TO 30
      MKEY=32-KEY
      IQ=MASK
      DO 20 I=1.MKEY
      JQ=IQ
   20 IQ=JQ/2
      IF(JQ.EQ.2*IQ) GO TO 40
   30 RETURN
   60 IF(MODE.EQ.1) CALL GCPFK(MALL.PFINT)
      IF(MODE.EQ.2) CALL  PFKP(MALL.PFINT)
      CALL SCOPLT(0.0.1.6.0.3.0)
      CALL SCOPLT(MODE.IDEV)
      IF(MODE.EQ.1) CALL GCPFK(MASK.PFINT)
      IF(MODE.EQ.2) CALL PFKP(MASK.PFINT)
   40 RETURN 1
   50 RETURN 2
      END
```

```
SUBROUTINE GREPLX(TEXT.NC)
DIMENSION TEXT(1)
CALL GREPLY(TEXT.NC)
CALL XBLANK(TEXT.NC)
RETURN
END
```

```
SUBROUTINE GRRPLX(TEXT,NC)
DIMENSION TEXT(1)
CALL GRRPLY(TEXT,NC)
CALL XBLANK(TEXT,NC)
RETURN
END
```

```
      SUBROUTINE MASKIT(K)
      EXTERNAL PFINT
      DIMENSION MMM(20)
      COMMON KEY,IOVLY,ITYPE,JTYPE(4),MASK
      DATA MMM/Z60000001,Z00000001,Z40000001,Z70000001,
     1         Z40000003,Z0000000D,Z60000003,Z7800000D,Z4000000D,
     2         Z00000003,Z5000000D,Z7C000001,Z70000003,Z0000000F,
     3        Z6000000F/
      MASK=MMM(K)
      CALL GCPFK(MASK,PFINT)
      RETURN
      END
```

```
      SUBROUTINE NEWPGE
      DIMENSION OUT(18,45),ALINE(1)
      DATA BLANK/' '/
      DATA EOL/Z40404015/
      DO 20 J=1,45
      DO 10 I=1,17
   10 OUT(I,J)=BLANK
   20 OUT(18,J)=EOL
      RETURN
      ENTRY STUFF(K,ALINE)
      DO 30 L=1,17
   30 OUT(L,K)=ALINE(L)
      RETURN
      ENTRY SNAP
      CALL GERAS(100)
      CALL GRDPLY(OUT,3240,&1000)
 1000 RETURN
      ENTRY PLUCK(K,ALINE)
      DO 40 LK=1,17
   40 ALINE(LK)=OUT(LK,K)
      RETURN
      ENTRY NEWLNE(K)
      DO 50 IK=1,17
   50 OUT(IK,K)=BLANK
      RETURN
      END
```

```
SUBROUTINE PASKIT(K)
EXTERNAL PFINT
DIMENSION MMM(20)
COMMON KEY.IOVLY.ITYPE.JTYPE(4).MASK
DATA MMM/Z00000001.Z60000001/
MASK=MMM(K)
CALL PFKP(MASK.PFINT)
RETURN
END
```

```
      SUBROUTINE PEWPGE
      DIMENSION OUT(18.45).ALINE(1)
      DATA BLANK/' '/
      DATA EOL/Z40404015/
      DO 20 J=1.45
      DO 10 I=1.17
   10 OUT(I.J)=BLANK
   20 OUT(18.J)=EOL
      RETURN
      ENTRY PSTUFF(K.ALINE)
      DO 30 L=1.17
   30 OUT(L.K)=ALINE(L)
      RETURN
      ENTRY PSNAP
      CALL REFILL
      CALL GRCHAR('BP'.OUT.3240.0..4000..IFLAG)
 1000 RETURN
      ENTRY PPLUCK(K.ALINE)
      DO 40 LK=1.17
   40 ALINE(LK)=OUT(LK.K)
      RETURN
      ENTRY PEWLNE(K)
      DO 50 IK=1.17
   50 OUT(IK.K)=BLANK
      RETURN
      END
```

```
      SUBROUTINE PWAIT(I,J,K,L)
      COMMON KEY,IOVLY,ITYPE,JTYPE(4)
      DATA MODE /2/
      JTYPE(1)=I
      JTYPE(2)=J
      JTYPE(3)=K
      JTYPE(4)=L
   10 CALL OWAIT
      WRITE(6,2000) ITYPE
 2000 FORMAT(20X,'ITYPE=',I8)
      CALL CHECK(&10,&20,MODE)
      RETURN
   20 CALL RLSEP
      CALL PLOT(0.,4.0,999)
      STOP
      END
```

```
      SUBROUTINE SPACE(N6)
      DO 10 I=1,N6
      CALL GRDPLY(' ',1,&1000)
   10 CONTINUE
 1000 RETURN
      END
```

```
      SUBROUTINE TWAIT(I,J,K,L)
      COMMON KEY,IOVLY,ITYPE,JTYPE(4)
      DATA MODE/1/
      JTYPE(1)=I
      JTYPE(2)=J
      JTYPE(3)=K
      JTYPE(4)=L
   10 CALL GWAIT
      CALL CHECK(&10,&20,MODE)
      RETURN
   20 CALL GRRLSE
      CALL PLOT(0.,4.0,999)
      STOP
      END
```

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* UNIVERSITY OF GEORGIA	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP UNCLASSIFIED

3. REPORT TITLE

(6) "GRAPHICAL AIDS FOR STATISTICAL COMPUTATION".

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

(9) Technical rept.

5. AUTHOR(S) *(Last name, first name, initial)*

(10) Walter P. Bond and Rolf E. Bargmann

6. REPORT DATE December, 1976 (11) Dec 76	7a. TOTAL NO. OF PAGES 336 (12) 343p.	7b. NO. OF REFS 31
8a. CONTRACT OR GRANT NO. N000 14-69-A-0423 (15) b. PROJECT NO. NR 042-261 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) Themis Report No. 37 (14) TR-112, THEMIS-UGA-37 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)* Report No. 112 Dept. of Statistics & Computer Science	

10. AVAILABILITY/LIMITATION NOTICES

Reproduction in whole or in part is permitted for any purpose of the United States Government.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

This report describes development and documentation of a set of programs for interactive use with graphics (IBM 360/370 with 2250 terminal), for the construction and evaluation of response surfaces, and for plotting and expanding statistical distribution functions. A facility is included to update and add new functions. Examples and sample displays are included for response surface optimization, process evaluation, likelihood function, power functions, multiple correlation distributions, Lelimann test, and distribution of the largest characteristic root.

405 576

DD FORM 1473 1 JAN 64

UNCLASSIFIED
Security Classification

KEY WORDS

Computer Graphics
Conversational Computation
Interactive Programs
Statistics
Distribution Functions
Response Surfaces
Interval Estimation
Extended Yates Algorithm
Splines
Contours